

by
M. J.
Winter

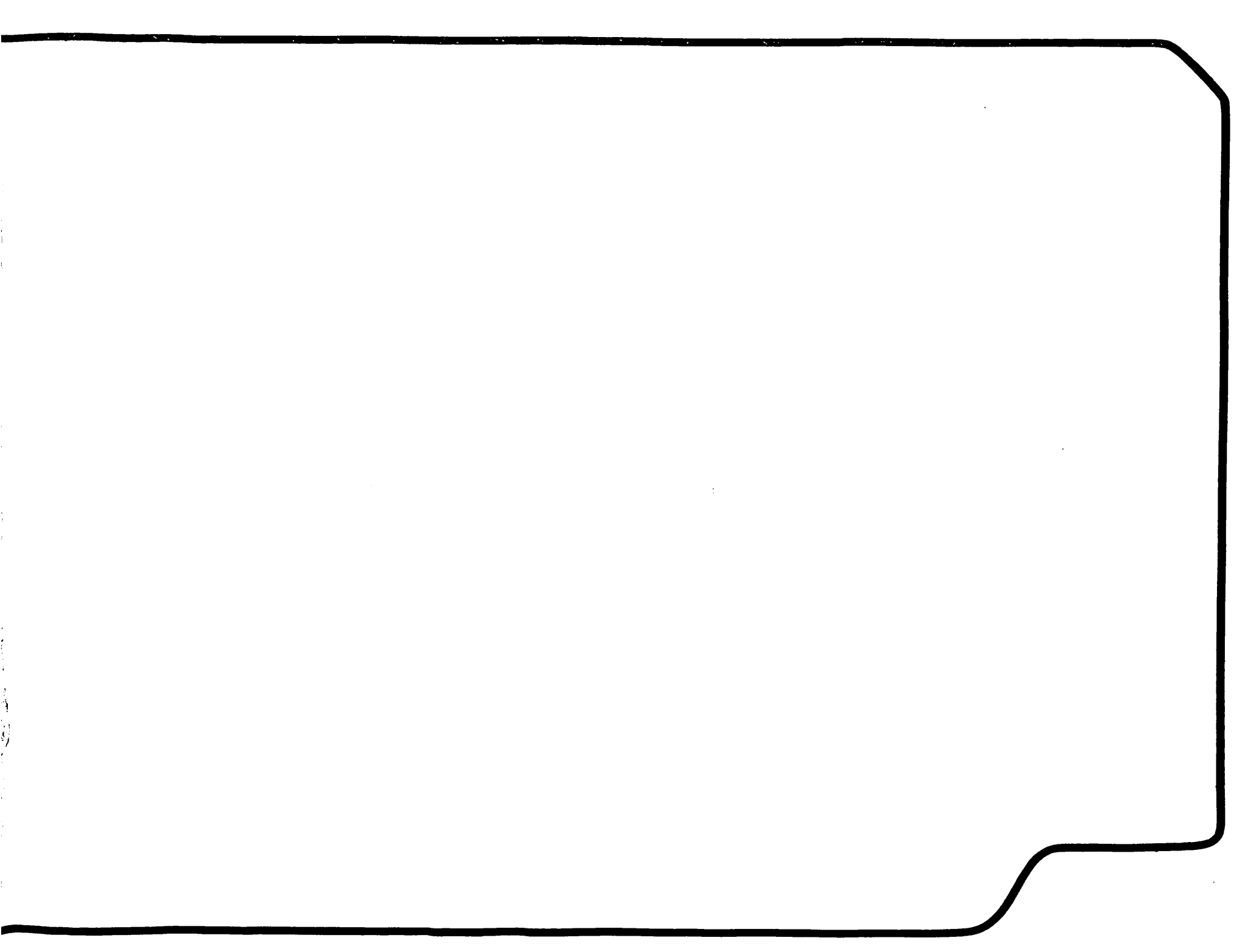
TI Version
by
Marcia
Carrozzo

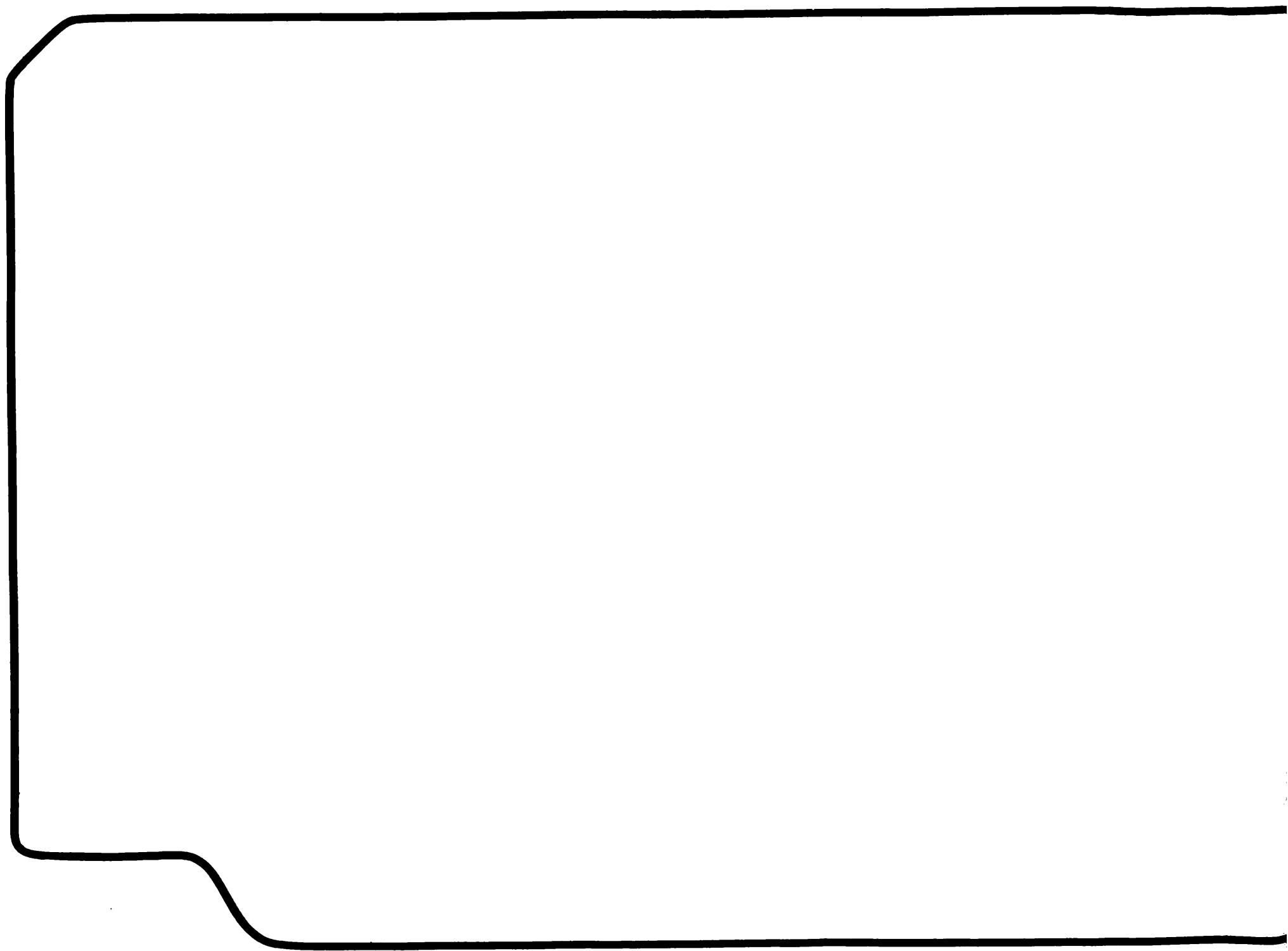
\$9.95



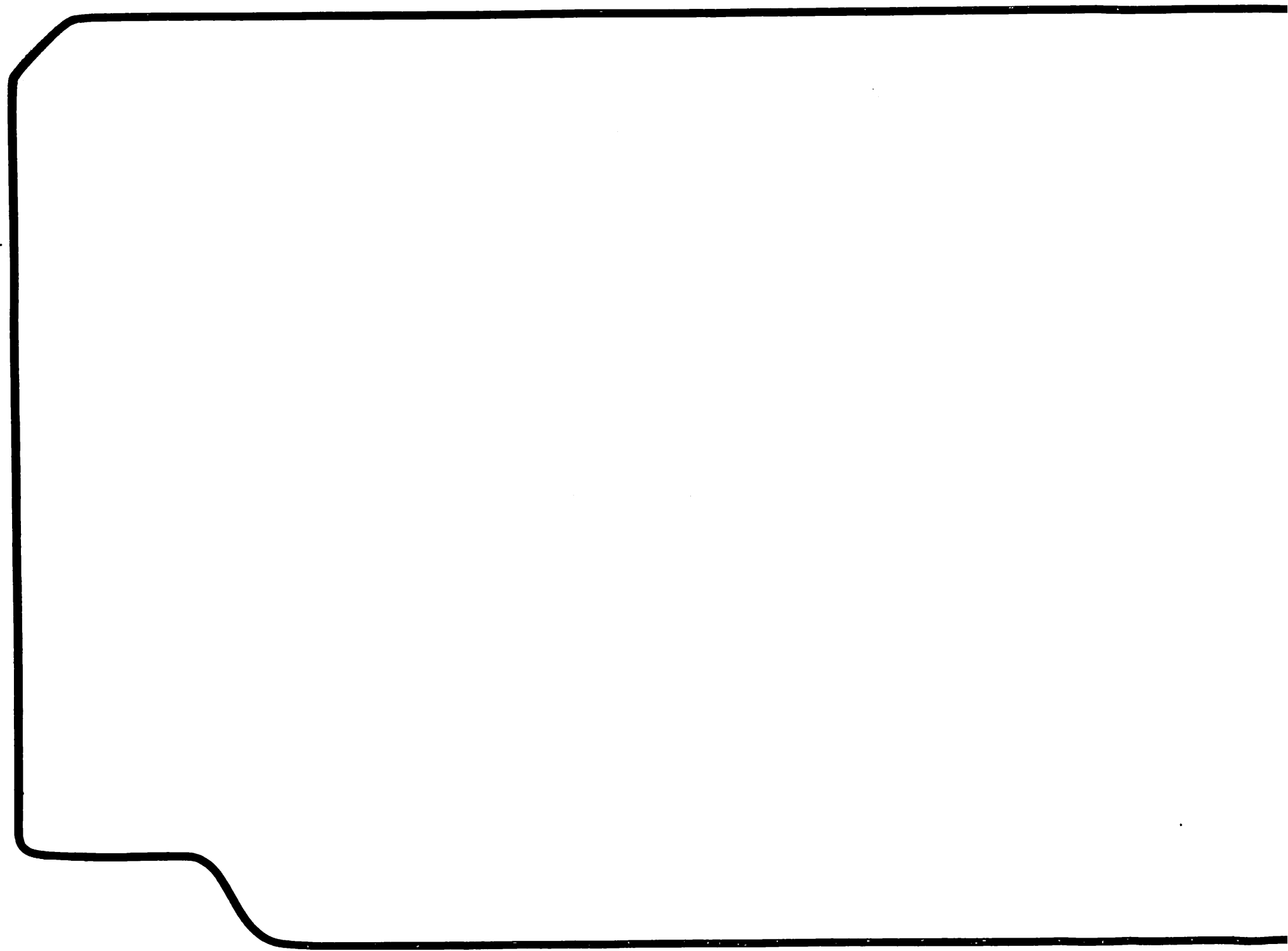
DATAMOST

TI 99/4A





Computer
Playground



Computer Playground

ON THE TI 99/4A

M.J. Winter, Ph.D.
Professor of Mathematics
Michigan State University

Illustrated by
Martin Cannon

TI Version by
Marcia Carrozzo

 **DATAMOST™**
8943 Fullbright Avenue
Chatsworth, CA 91311-2750
(213) 709-1202



This manual is published and copyrighted by DATAMOST INC. Copyright, duplicating, selling or otherwise distributing this product is hereby expressly forbidden except by written consent of DATAMOST, INC.

The word TI and TI logo are registered trademarks of Texas Instruments Incorporated.

Texas Instruments Inc. was not in any way involved in the writing or other preparation of this manual, nor were the facts presented here reviewed for accuracy by that company. Use of the term TI should not be construed to represent any endorsement, official or otherwise, by Texas Instruments Incorporated.

Copyright 1983 Datamost Inc.

ACKNOWLEDGEMENTS

My thanks go to all those who encouraged me to write something for younger children. Special appreciation is due to:

Dave Gordon and his very creative and supportive staff at DATAMOST.

My fellow instructors at the MSU summer computer camps: Ed Carlson, John Forsythe and Mark Lardie. Double thanks to Ed with whom I wrote materials for the camp, and who encouraged this project.

The children in the first classes; especially Amy, Linda, Harriet, Claire, Ida, Elizabeth, Beth, Susie, Leah, Jason, Tammi, Courtney, Jessica, Erin and Jon.

All the teachers who've responded to my articles and programs describing the enthusiasm of their students.

My family, who not only live with my three computers in the kitchen, but who liberally gave me suggestions and encouragement.

The 6-year old, now 8, who started this project off when she complained:

I don't want to just play games.
Show me how to make the computer DO something.

THE PROBLEMS

1. Season's Greetings
2. Take a Number, Please
3. Mistakes
4. Initials
5. Where Now?
6. Wallpaper
7. Today's Message
8. Screen Color
9. Sound Off
10. Letter to Santa
11. Pausing for Breath
12. Rockets
13. Counting
14. Touch Your Toes Ten Times
15. Movers
16. The Worm
17. More Counting
18. Testing with IF...THEN
19. Goldilocks
20. Animal Guess
21. Little Indians
22. Nursery Rhyme
23. Blinking Lizards
24. PEEKaboo
25. Funnybones and Sore Toes
26. Quiz Time
27. Pieces of Words
28. Sound and Light Show
29. Rhyming Game
30. Compound Words
31. Word Puzzle
32. Doing Math
33. Math Quiz
34. Times Tables
35. Days and Nights
36. Countdown
37. Think of a Number
38. Two Dice
39. Get Out of Jail
40. Harder Math Quiz
41. Number Guess
42. Answer Machine
43. Dinner Darts
44. Horoscope

INTRODUCTION

This book is for the child who wants to make the computer “do” something. It is based on the perception that such a child is probably not thinking of evaluating arithmetic expressions, but rather of making the computer respond to his/her directions. A child who knows some BASIC will be able to “talk” to the computer and control its actions.

The BASIC taught in this book will enable children to make the computer “do” things which interest them. The method of presentation is that of “conversational BASIC”: each idea is taught by the combination of an interesting format and lots of repetition.

The book is intended for the child in grades 2.5 - 7 whose family has a TI computer, or for children who have access to a classroom computer. While the programs presented are specifically designed to work on the TI, this is not a book on “What You Can Do with Your TI.” Rather, it is a book on how to use BASIC with an TI. While many of the special TI features are included, they are used only as illustrations or applications of BASIC words.

Each activity is presented as a “Problem” in a workbook. Children will find the problems both interesting and geared to their level of understanding. They will feel satisfaction in being able to fill in the blanks in the workbook. As the book is completed, a stock of successful programs is accumulated. Frequently a child will spend long periods of time repeating and varying a single activity.

OVERVIEW

Children are able to mimic before they can create. In some of the problems they are directed to run sample programs, then modify them. In others, the children are to complete partially written programs.

Most young children will need a HELPER available (parent, teacher, older child). The helper should read the COMMENTS AND DESCRIPTIONS section to see what is required and what explanations are suggested. Older children may need occasional assistance.

The BASIC vocabulary of each group of problems is given below. See the COMMENTS AND DESCRIPTIONS section for more details.

Group	Problems	BASIC Vocabulary
1	1 - 5	PRINT, GOTO, END
2	6 - 10	INPUT, CALL CLEAR, CALL COLOR, CALL SOUND
3	11 - 17	FOR...NEXT
4	18 - 23	IF...THEN
5	24 - 25	CALL KEY
6	26 - 31	SEG\$, LEN
7	32 - 36	arithmetic operations

8	37 - 41	Randomize, INT(6*RND)+1
9	42 - 44	DIM A\$(10)

COMMENTS AND DESCRIPTIONS

Problems 1 - 5

The first problems teach the words PRINT and GOTO, as well as the purpose of line numbers.

1. **SEASON'S GREETINGS** The program prints an endless number of greetings. To stop it, press FCTN 4. PRINT by itself makes the computer skip a line.
2. **TAKE A NUMBER, PLEASE** The line numbers attached to each statement determine the order in which the computer goes through the program. Even if the lines are entered in mixed-up order, the computer knows which should come next. Type LIST to see the lines in their correct order.
3. **MISTAKES** The message * INCORRECT MESSAGE is likely to mean something is misspelled (a typing error) or you probably forgot the quotation marks.
4. **INITIALS** The example makes a large P from small ones. To design a slanted initial, like A or W, it may be helpful to use a grid.

5. **WHERE NOW?** A puzzle which emphasizes that the computer will always go to the next higher line number if it isn't told to go somewhere else. END is necessary to stop the program without printing extra letters. A semicolon at the end of a PRINT statement keeps the letters all printed on the same row.

Problems 6 - 10

Most children have seen a computer ask "what's your name?". The INPUT statement makes the computer stop until a word is typed and RETURN is pressed. CALL COLOR is used to change the color of the screen and the message. CALL SOUND creates a variety of musical tones and noises.

6. **WALLPAPER** The computer prepares a "box" labeled P\$. When the pattern is entered, the computer stores it in the box. Every time the program needs the pattern, it goes to the box and looks inside. (Demonstrate with a labeled box and a piece of paper with a pattern on it.)
7. **TODAY'S MESSAGE** This program reinforces the idea of a variable. Three different variables are entered, then put together in a sentence.
8. **SCREEN COLOR** The computer will print in color if CALL COLOR is used. You can change the screen color too.

9. **SOUND OFF** Experiment with single notes and then try playing up to four notes at the same time. Noises are made by using a - (minus) sign before the number.
10. **LETTER TO SANTA** (Young children will object on the grounds that they're too old for this.) Four words are used in this program. A matching exercise has the child connect the name on each "box" with what it contains.

Problems 11 - 17

The next group is based on the FOR . . . NEXT instructions. The first instance of FOR . . . NEXT is to make the computer pause. In other applications it does something each time it counts. Cursor controls are used in graphics applications of loops and delays.

11. **PAUSING FOR BREATH** The computer prints a line, then pauses while it counts to 18000, then prints another line. Use a stopwatch or watch to time how fast it can count.
12. **ROCKETS** The pause and special effects are combined in this program which sends rockets into the blue sky.
13. **COUNTING** Watch the computer while it counts. How much longer does it take to count to 18000 when the numbers are printed? Finally, a reaction test checks our reflexes.

14. **TOUCH YOUR TOES . . .** Make the computer repeat an action while it counts. It can print a message, skip lines or draw the sides of a box.
15. **MOVERS** A double application of FOR . . . NEXT. One FOR . . . NEXT is used to repeat the printing of the star; the other is used to slow down the printing. Movement is really an illusion. If a star is printed, erased and printed again one space further along, it will appear to move.
16. **THE WORM** The same as #15, but this time a worm crawls across the screen. To erase the tail, it is necessary to erase and print the characters in new locations.
17. **MORE COUNTING** A review of FOR . . . NEXT. This time the upper limit is an INPUT. When INPUT is asking for a number, the \$ is not used.

Problems 18 - 23

These problems introduce the idea of testing, i.e., comparing two things. Testing is done with IF something is true THEN do something. The equals sign, =, means "is the same as"; the sign "<>" means "is not the same as." Testing is done first with words, then with numbers.

18. **TESTING** Simple and complicated statements can be tested. IF A\$="RED" or A\$="BLUE" is an example. THEN is followed by a line number where an action (usually PRINT) takes place.
19. **GOLDILOCKS** Complete the program by testing the temperature of the porridge.
20. **ANIMAL GUESS** This program uses IF . . . THEN with branching. Younger children will probably be content with changing line 100 every time. Others may want to make this a two-person game.
21. **LITTLE INDIANS** A short program combines FOR . . . NEXT to print out the counting rhyme. This is more complicated than it appears as the child must supply the limits and the final message. GOSUBs may be used for some of the PRINT statements.
22. **NURSERY RHYME** The problem is to write a program which prints out a rhyme like that in #19.
23. **BLINKING LIZARDS** Make the orphan blink her eyes. To make only the eyes change, we print on top of them.

Problems 24 - 25

These problems use CALL KEY. CALL KEY is used to look at what's in a certain memory location. The location dealt with contains information about the name of the last key touched. All of this information is in special codes.

24. **PEEKABOO** While a program is running, the computer constantly checks the keyboard to see what key is being pressed. If it's FCTN 4, the program will stop. Each time a new key is pressed, the code number for that key goes into location K. Even though we take our hands away, the number in doesn't change until a different key is pressed.
25. **FUNNYBONES...** Use the code number found in #24 to make the computer respond differently to certain keys.

Problems 26 - 31

These problems are concerned with words and parts of words. SEG\$ and LEN defined. Adding words (concatenation) builds new words from old ones.

26. **QUIZ TIME** While no new BASIC concepts are introduced, this is more complicated than any of the earlier programs. All the answers are stored in the A\$ box. Answers are checked; sometimes a wrong answer produces another chance.
27. **PIECES OF WORDS** An investigation into SEG\$(A\$,1,1) and LEN(A\$). Some children need help seeing that LEN(A\$) is the number of letters in A\$.
28. **SOUND AND LIGHT SHOW** When completed, this program will play random notes while flashing random screen colors; the rhythm is also random.
29. **RHYMING GAME** String functions and LEN(A\$) are used to give hints to the guesser. Children will enjoy thinking of tricky rhymes that aren't given away by the two hints.
30. **COMPOUND WORDS** New words can be formed by gluing together beginnings and endings of other words. A word can be glued to itself to make it twice as long.
31. **WORD PUZZLE** Using the beginnings and endings of words, print out other words. This should be done with a pencil and then checked.

Problems 32 - 36

Very simple arithmetic operations on variables are introduced in these problems. Experience has shown that manipulating variables generally requires a high level of mental development and the ability to handle abstractions. The operations in these problems are kept as concrete as possible. Counting backwards is presented here.

32. **DOING MATH** The operation is doubling. The * is the multiplication sign on the computer. Change the operation to tripling. Have older children change the program to dividing by 2, doubling and adding 3, etc.
33. **MATH QUIZ** Two values are input. The user is asked to add them; the computer checks the answer. An extension would be to have a wrong answer cause the problem to be printed again. All children can change this to a multiplication quiz.
34. **TIMES TABLES** First the computer prints multiples of 2. Then labels are added so that the output looks like a multiplication table. Younger children may have difficulty understanding the elaborate PRINT statement, but they will be delighted with the output and will be able to change it.

35. **DAYS AND NIGHTS** Use the computer to solve the problem of how much we sleep in a year if we sleep eight hours a night. Several variables are used, but their names are highly mnemonic.
36. **COUNTDOWN** Simple demonstrations of how to count from 5 down to 1.

Problems 37 - 41

This group of problems is centered around making a program "think" of a number. This is done by introducing the line

RANDOMIZE

and later referring to $\text{INT}(6*\text{RND})+1$ or $\text{INT}(8*\text{RND})+1$. Line 10 is intended as a magic formula; no explanation is given of why it works. Older children (11+) will be able to follow the explanation:

RND is a (decimal) number between 0 and 1

$6*\text{RND}$ is a (decimal) number between 0 and 6

$6*\text{RND}+1$ is a (decimal) number between 1 and 7

INT means take the whole number part, so

$\text{INT}(6*\text{RND})+1$ is one of: 1, 2, 3, 4, 5, 6

37. **THINK OF A NUMBER** How to use the formula to roll a die. The results of six rolls are recorded.
38. **TWO DICE** Every time the computer has to think of a number from 1 to 6, it must refer to $\text{INT}(6*\text{RND})+1$. The lines of this program are used in the next problem, so don't NEW it.
39. **GET OUT OF JAIL** The classical reason for rolling two dice. The program uses IF . . . THEN for the test. Remind the child that "=" means "is the same as."
40. **HARDER MATH QUIZ** Use the number-thinker to create the math problems. This usually results in a more difficult quiz.
41. **NUMBER GUESS** The computer thinks of a number and the user tries to guess it. Messages of TOO BIG and TOO SMALL are printed. A counter is added to count how many guesses are used. Compare it with the highway counters: every time a car drives over it, it increases the counter.

Problems 42 - 44

Although the concept of an array is abstract, the idea of a list of words is not. The computer must know how many words can be in the list; `DIM A$(10)` means at most 10 words in the list. The first word is `A$(1)`, the second is `A$(2)`, etc.

In these problems `INT(X*RND)+1` is used to grab a word from the list. The magic line 10 is always used.

- 42. **ANSWER MACHINE** Like a crystal ball, the computer can see into the future. After being asked a question, the computer randomly selects one of its supply of answers.
- 43. **DINNER DARTS** Throw the dart to see whether dinner is hot dogs or steak.
- 44. **HOROSCOPES** The computer turns fortune-teller by selecting words from three different lists. Each selection requires `INT(X*RND)+1`. To get all possibilities on the lists, attention must be paid to the `INT(?*RND)+1`.

PROBLEM 1 SEASON'S GREETING

A. Enter this program - (that means after typing each line, you press ENTER) - and RUN it.

```
10 PRINT "HAPPY"  
20 PRINT "THANKSGIVING"
```

Copy what happened _____



B. Enter two more lines:

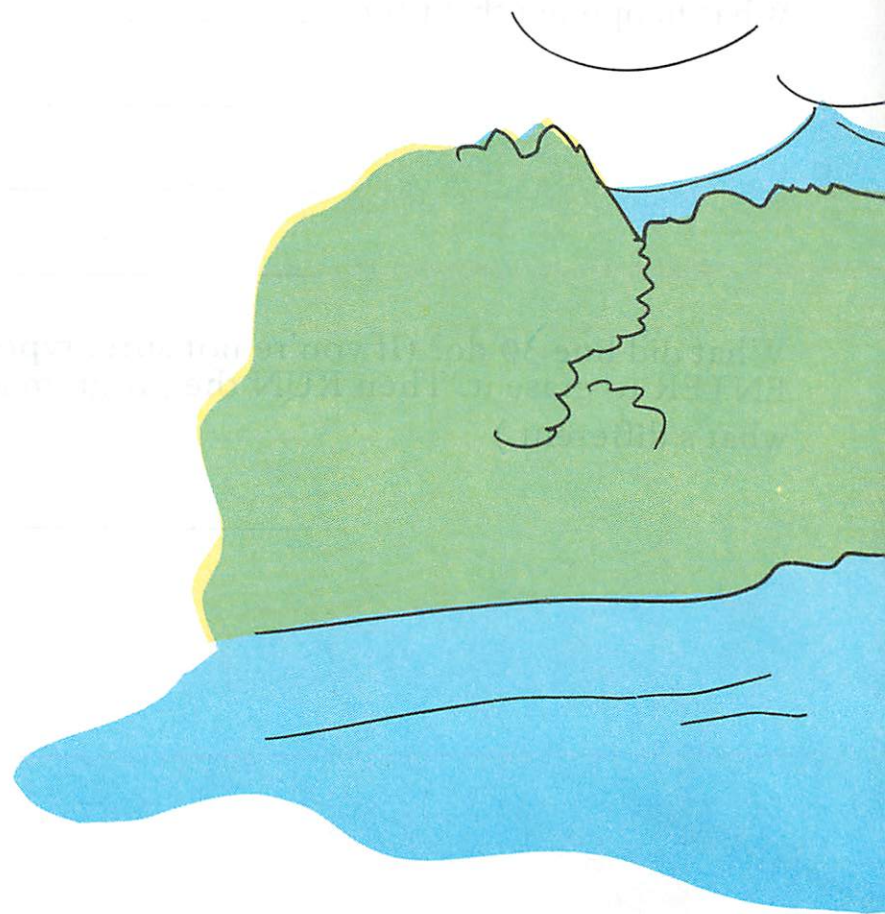
```
30 PRINT  
40 GOTO 10
```

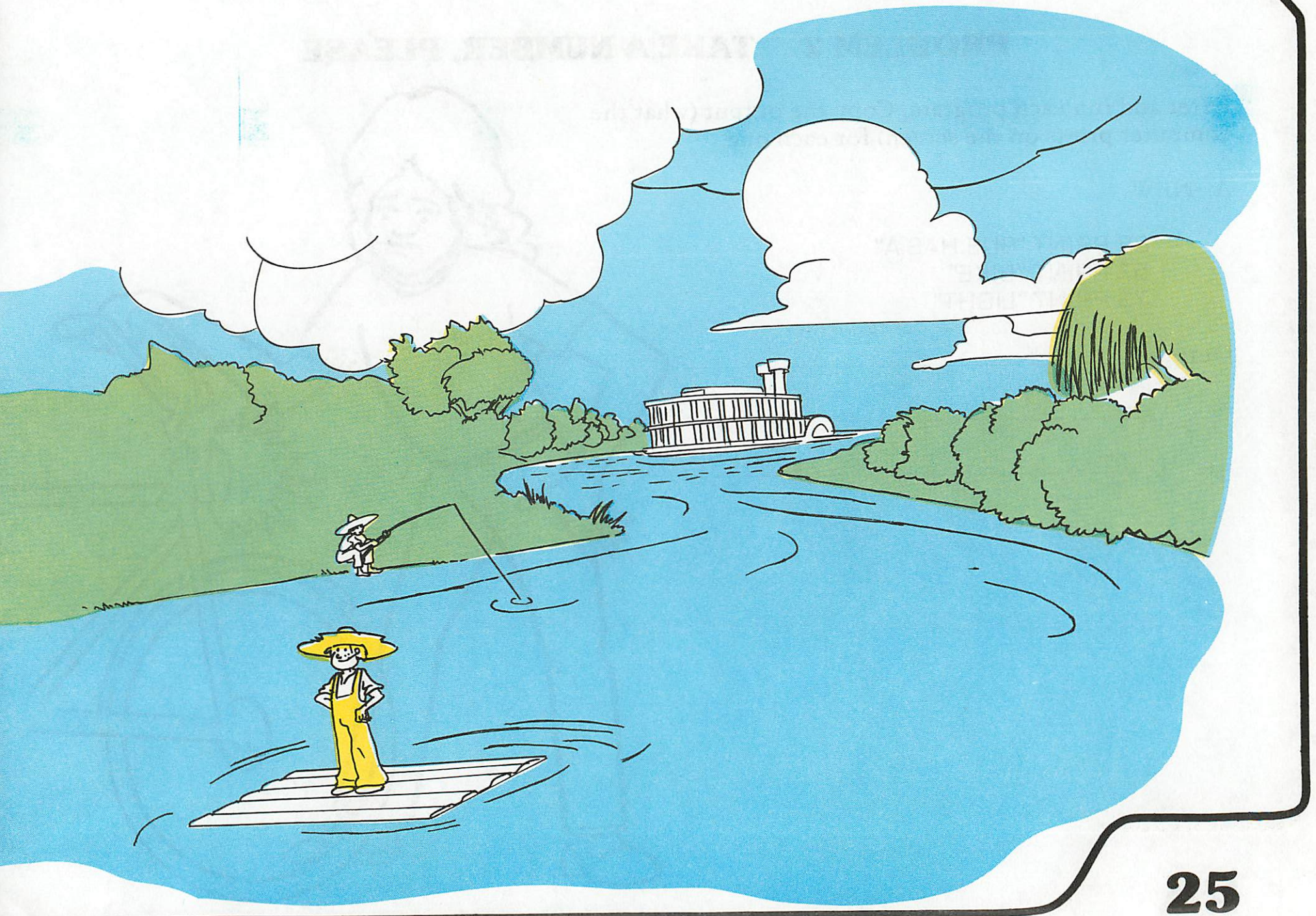
and RUN the program again. To stop it, press FCTN 4 (hold down FCTN and press 4).

What happened this time? _____

What did line 30 do? (If you're not sure, type 30 and press ENTER to erase it. Then RUN the program again and see what's different.)







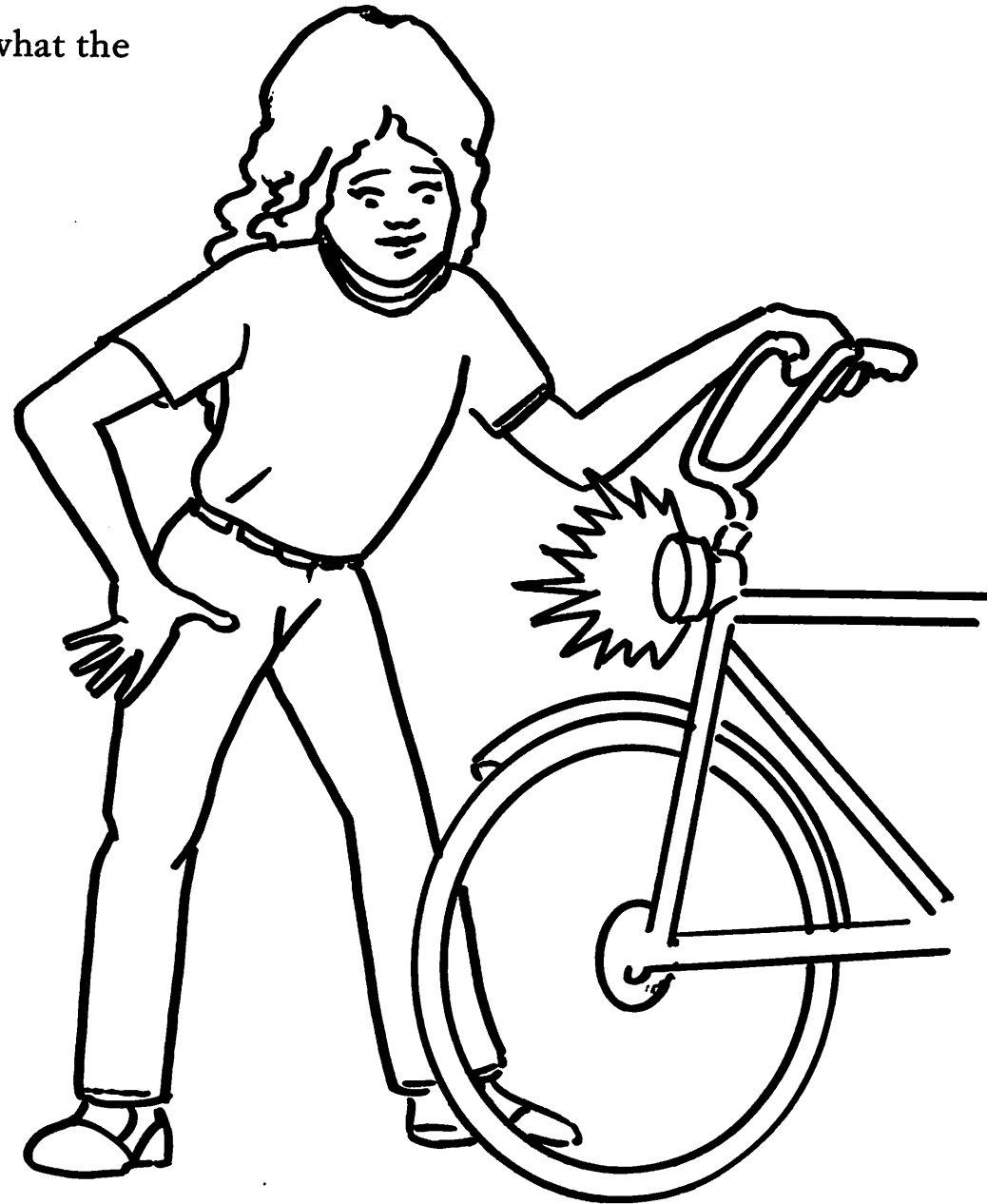
PROBLEM 2 TAKE A NUMBER, PLEASE

Enter and run each program. Copy the output (what the computer prints on the screen) for each one.

A. NEW

```
38 PRINT "SUE HAS A"  
57 PRINT "BIKE"  
73 PRINT "LIGHT"
```

Output _____

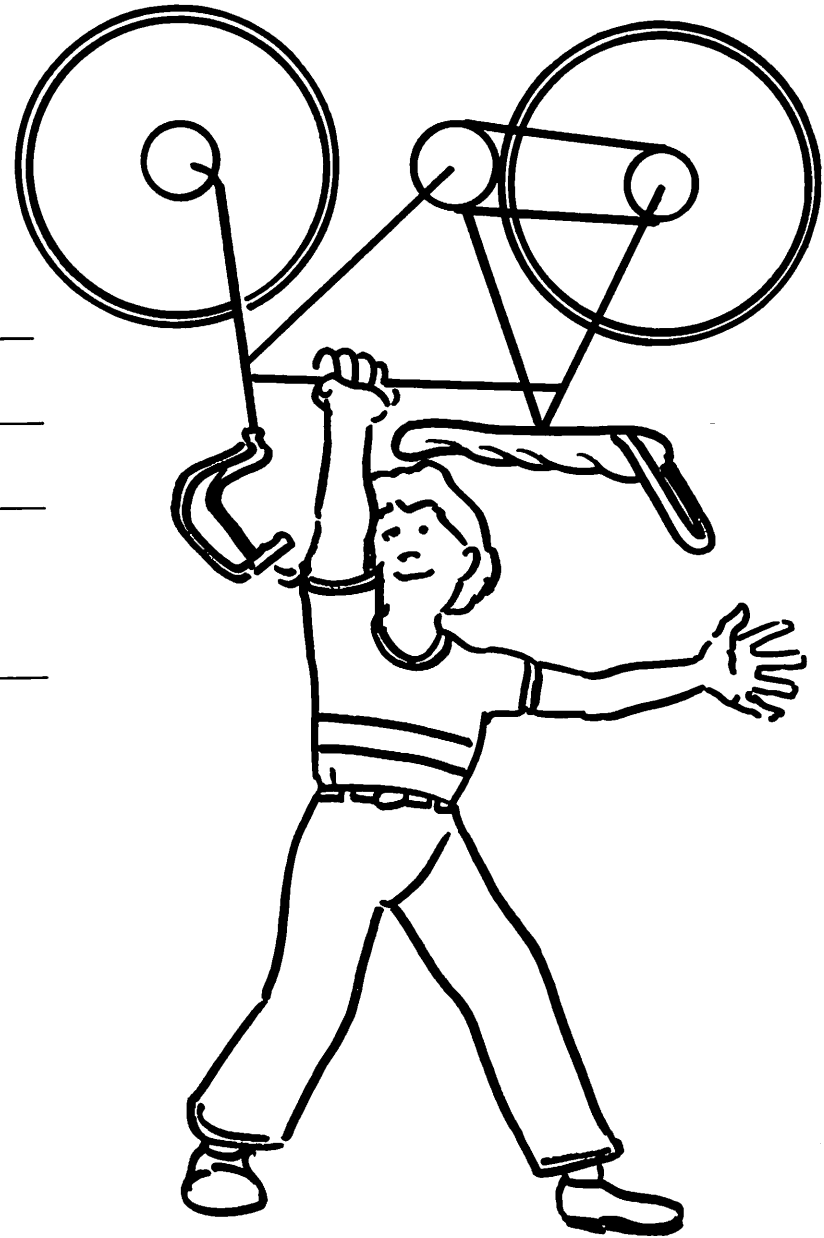


B. NEW

```
86 PRINT "SAM HAS A"  
93 PRINT "BIKE"  
88 PRINT "LIGHT"
```

Output _____

Why did the programs print something different?



PROBLEM 3 MISTAKES

Enter this program, EXACTLY as it appears here:

```
10 PRINT "RED"  
20 PRINT "HOT"  
30 PRINT PEPPERS
```

Run it.

What happens? _____

Fix line 20.

20 _____

Run the program now.

What happens? _____



Change line 30 so that the program prints:

RED
HOT
PEPPERS

30 _____



PROBLEM 4 INITIALS

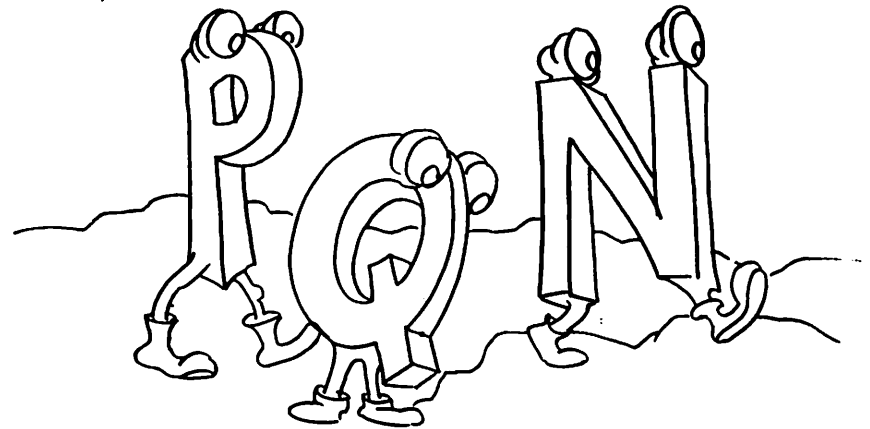
Run this one:

```
10 PRINT
20 PRINT
30 PRINT "PPPP"
40 PRINT "P  P"
50 PRINT "PPPP"
60 PRINT "P "
70 PRINT "P "
80 PRINT "P "
```

Write a program like this that uses your initial; it should make a large initial out of several small ones.

Copy your program here: (You might not need all the lines.)

```
10 _____
20 _____
30 _____
40 _____
50 _____
```

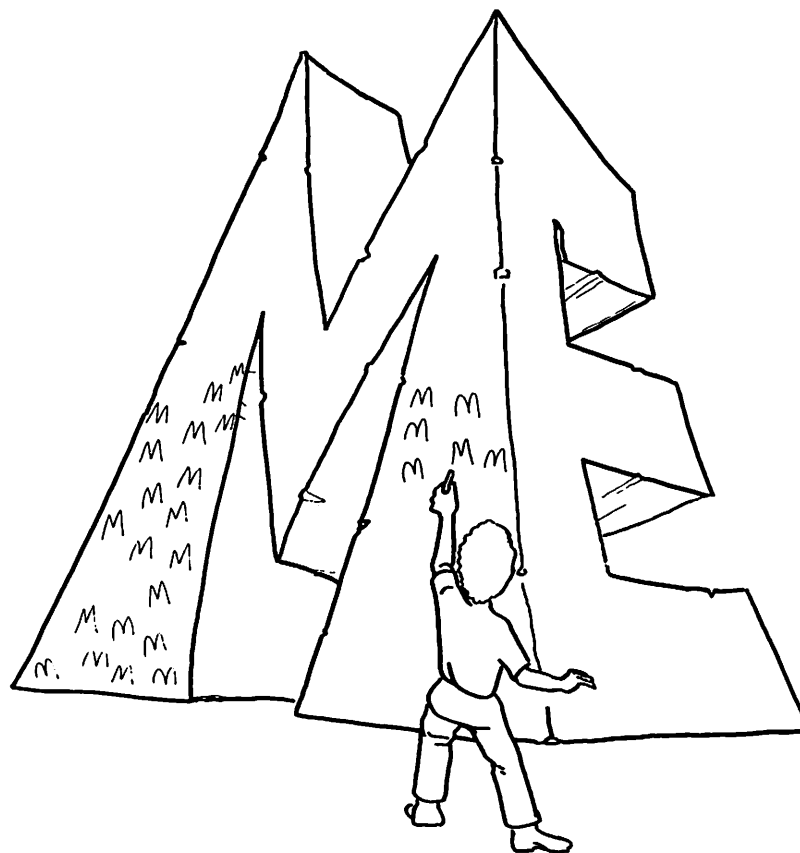


60 _____

70 _____

80 _____

90 _____



PROBLEM 5 WHERE NOW?

Unless you tell it to go somewhere else, a program which just finished one line will go on to the next line number.

A. Look at this program. PREDICT what it will do. Then enter and run it to check.

```
10 GOTO 35  
15 PRINT "T";  
20 GOTO 45  
25 PRINT "A";  
30 GOTO 15  
35 PRINT "R";  
40 GOTO 25  
45 END
```

Prediction: _____



B. Fill in the GOTO statements to make the program print ART.

10 GOTO _____

15 PRINT "T";

20 GOTO _____

25 PRINT "A";

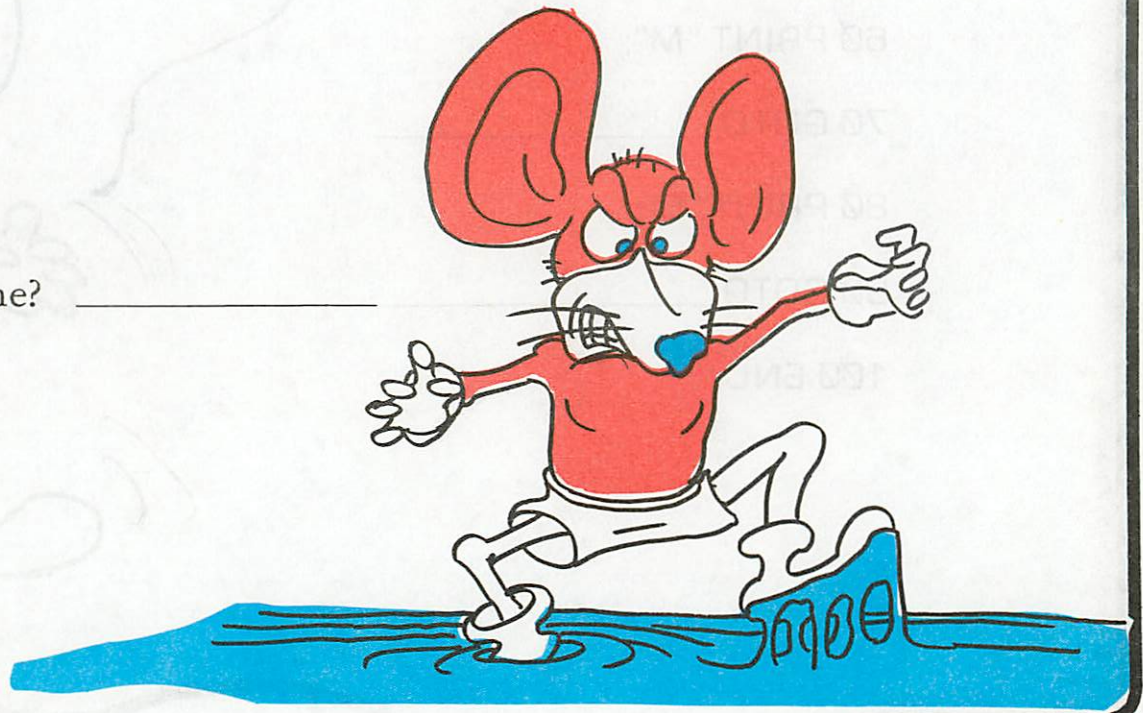
30 GOTO _____

35 PRINT "R";

40 GOTO _____

45 END

What makes the letters print on the same line? _____



C. Complete this program so it prints MICE.

10 GOTO _____

20 PRINT "E";

30 GOTO _____

40 PRINT "I";

50 GOTO _____

60 PRINT "M";

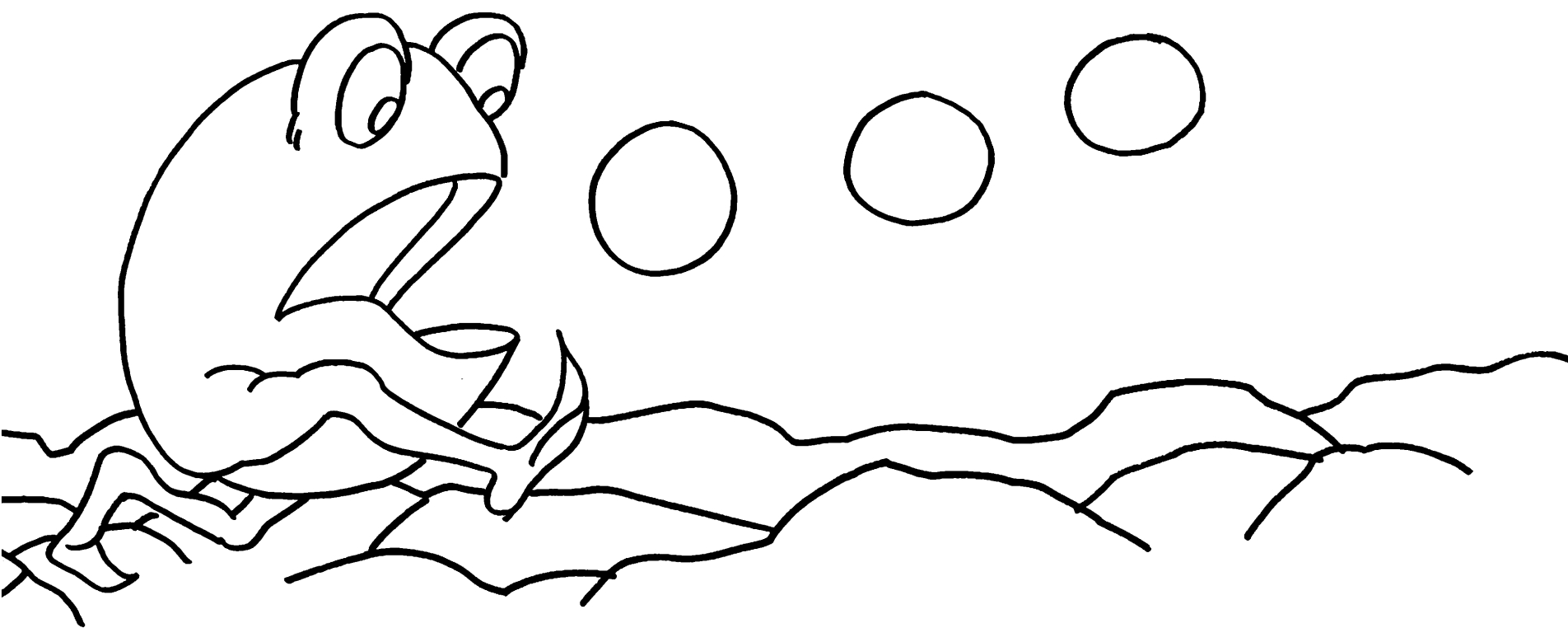
70 GOTO _____

80 PRINT "C";

90 GOTO _____

100 END





PROBLEM 6 WALLPAPER

This program will design wallpaper. You'll give it a basic pattern, and it will keep repeating it.

Each time it's run, the program will ask for the pattern. When it does, type in a pattern and press RETURN.

The computer will put the pattern in a special place, called P\$. Each time the program comes to line 40, it looks in P\$ to see what the pattern is.

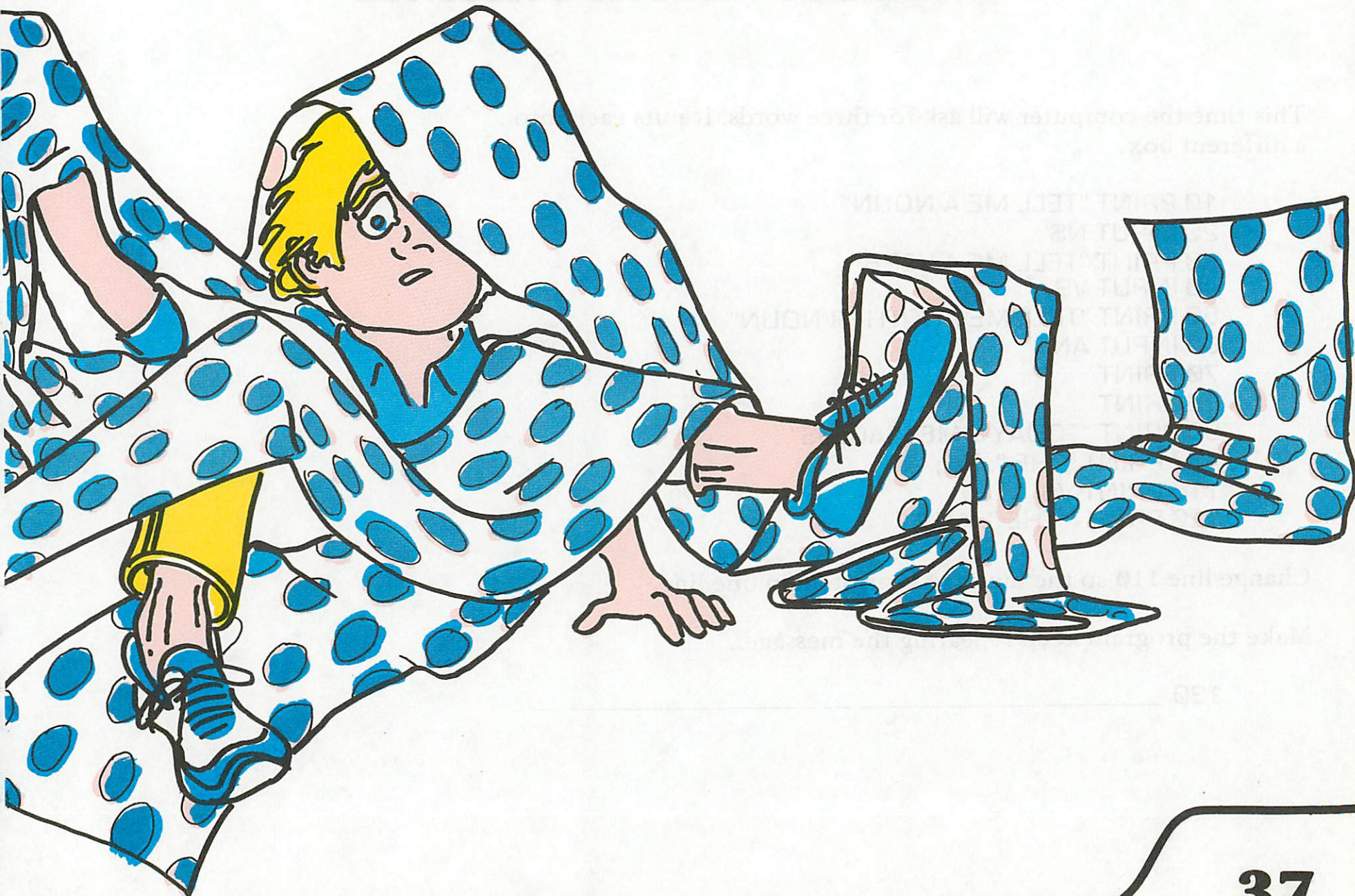
```
10 PRINT "WHAT'S THE PATTERN"  
20 INPUT P$  
30 PRINT P$;  
40 GOTO 30
```

Try patterns like these:

* *

A . . . V . . .





PROBLEM 7 TODAY'S MESSAGE

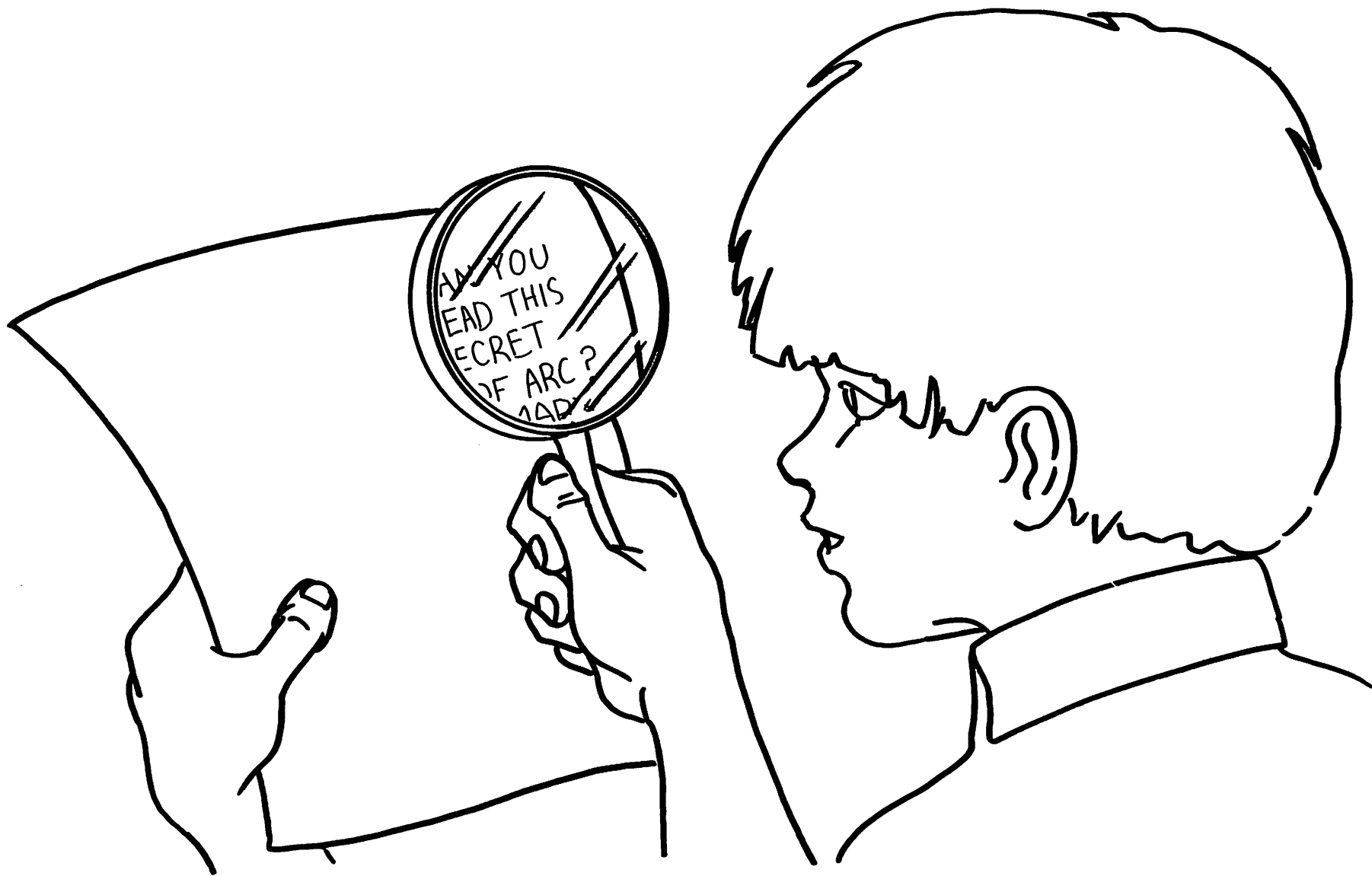
This time the computer will ask for three words. It puts each into a different box.

```
10 PRINT "TELL ME A NOUN"  
20 INPUT N$  
30 PRINT "TELL ME A VERB"  
40 INPUT V$  
50 PRINT "TELL ME ANOTHER NOUN"  
60 INPUT AN$  
70 PRINT  
80 PRINT  
90 PRINT "TODAY'S MESSAGE IS"  
100 PRINT "THE "; N$;" ";  
110 PRINT V$  
120 PRINT " THE "; AN$
```

Change line 110 so the message is printed on one line.

Make the program keep repeating the message.

```
130 _____
```



PROBLEM 8 SCREEN COLOR

You can make the TI print in color on different colored screens by using the commands CALL SCREEN and CALL COLOR.

Try this:

```
100 CALL CLEAR
120 INPUT "WHAT COLOR FOR THE CHARACTER":C
130 INPUT "WHAT COLOR FOR THE SCREEN":S
140 CALL SCREEN(S)
150 CALL COLOR(2,C,S)
160 CALL VCHAR(1,12,42,24)
```

Try different color combinations using numbers 1 - 16.





PROBLEM 9 SOUND OFF

On the TI there is a BASIC command called CALL SOUND. You can play up to three musical notes and one "noise" at the same time.

This program will make a single, short sound:

```
10 PRINT "ENTER NOTE NUMBER"  
20 INPUT N  
90 CALL SOUND(1000,N,1)
```

Enter and run this program to see how it works. Try these values for N: 349, 220, 262, 330.

To play two notes at the same time, add these lines and change line 90.

```
30 PRINT "ENTER SECOND NOTE NUMBER"  
40 INPUT S  
90 CALL SOUND(1000,N,1,S,1)
```

Add a third note by finishing these two lines and changing line 90:

```
50 PRINT _____  
60 INPUT _____
```

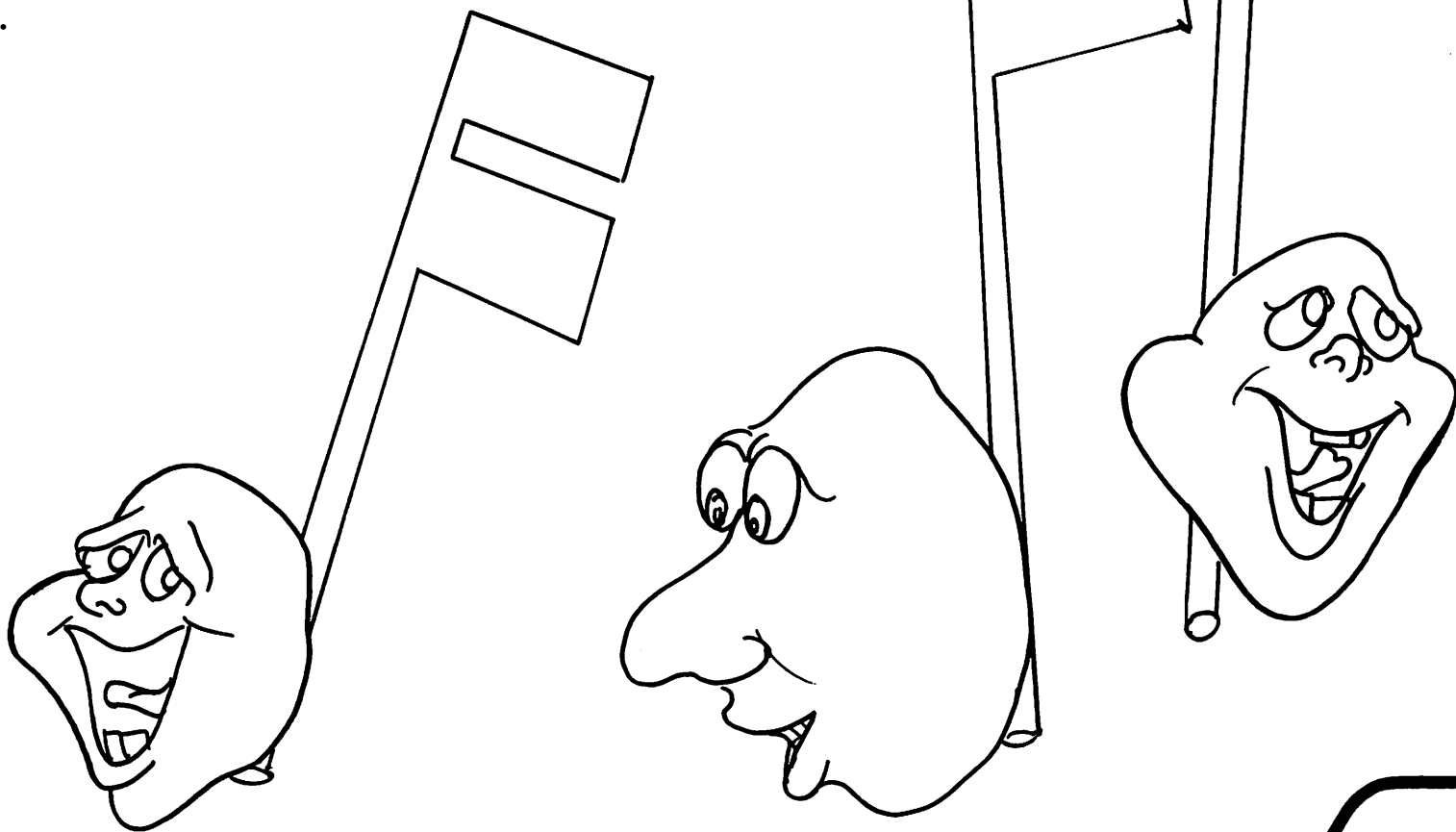


A noise is made by using a minus sign(-) in front of the numbers 1 - 8. Add a noise to your program by completing the next two lines. Remember to change line 90.

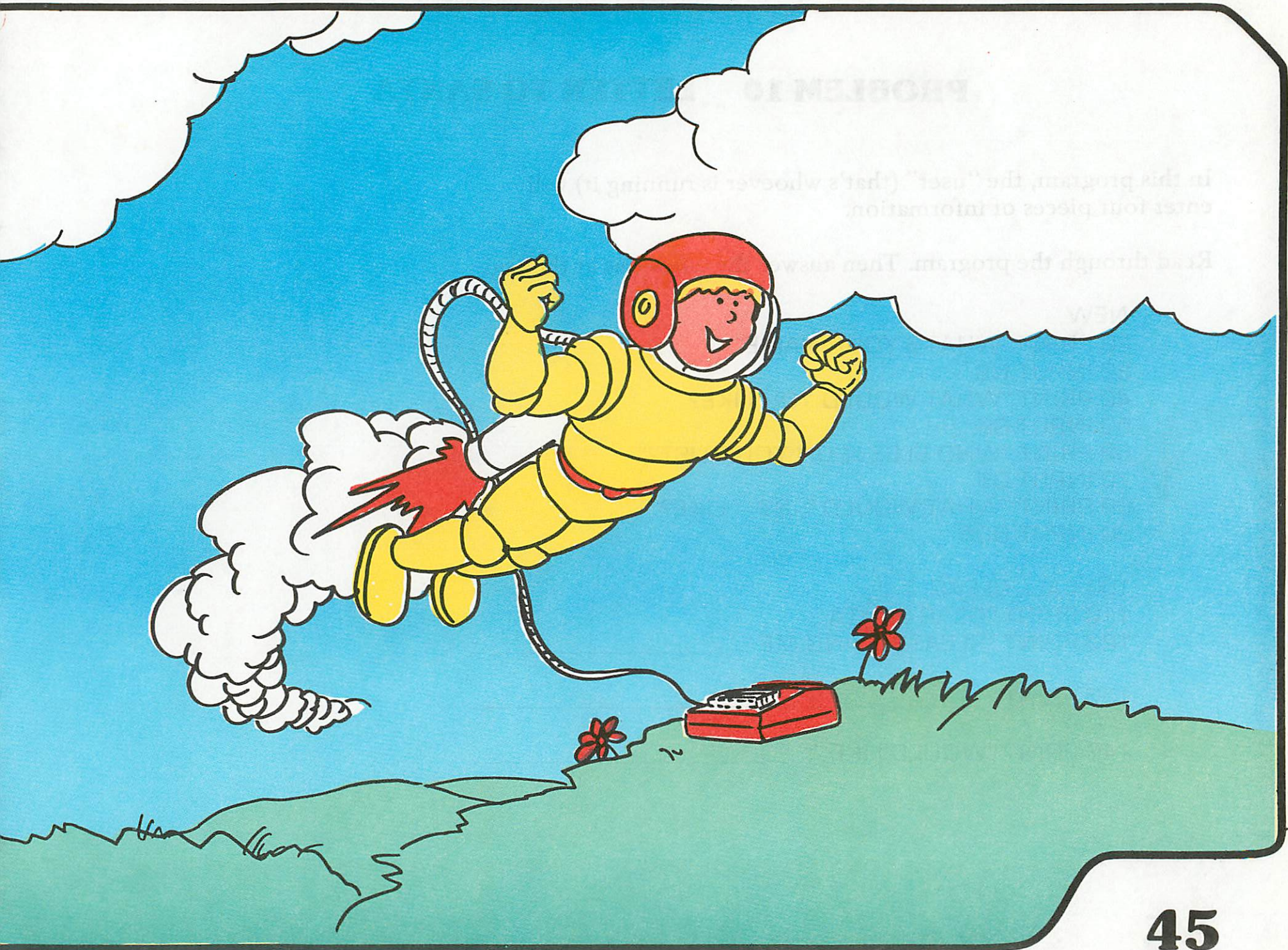
70 PRINT _____

80 INPUT _____

Change the 1000 in line 90, first to make the sounds shorter and then longer.







PROBLEM 10 LETTER TO SANTA

In this program, the “user” (that’s whoever is running it) will enter four pieces of information.

Read through the program. Then answer the questions at the end.

```
NEW
20 PRINT "WHAT'S YOUR NAME"
30 INPUT N$
40 PRINT "WHAT WOULD YOU LIKE?"
50 INPUT G$
60 PRINT "WHO ELSE IS ON YOUR LIST?"
70 INPUT Q$
80 PRINT "WHAT WOULD "; Q$; " LIKE?"
90 INPUT QG$

100 CALL CLEAR
110 PRINT "DEAR SANTA,"
120 PRINT "PLEASE BRING ME "; _____

130 PRINT _____

140 PRINT "WOULD LIKE "; _____
```

```
150 PRINT "THANK YOU."  
160 PRINT  
170 PRINT "LOVE"
```

```
180 PRINT _____
```

Draw a line from each "name" to what it stands for.

N\$	gift you'd like
G\$	other person's name
Q\$	other person's gift
QG\$	your name

One of the "names" should be written in each set of
_____ in the program.

Put in the names and run the program to check it.



PROBLEM 11 PAUSING FOR BREATH

To slow down the computer, we can make it stop and count to itself. Since it counts very fast, we'll have to make it count to a high number.

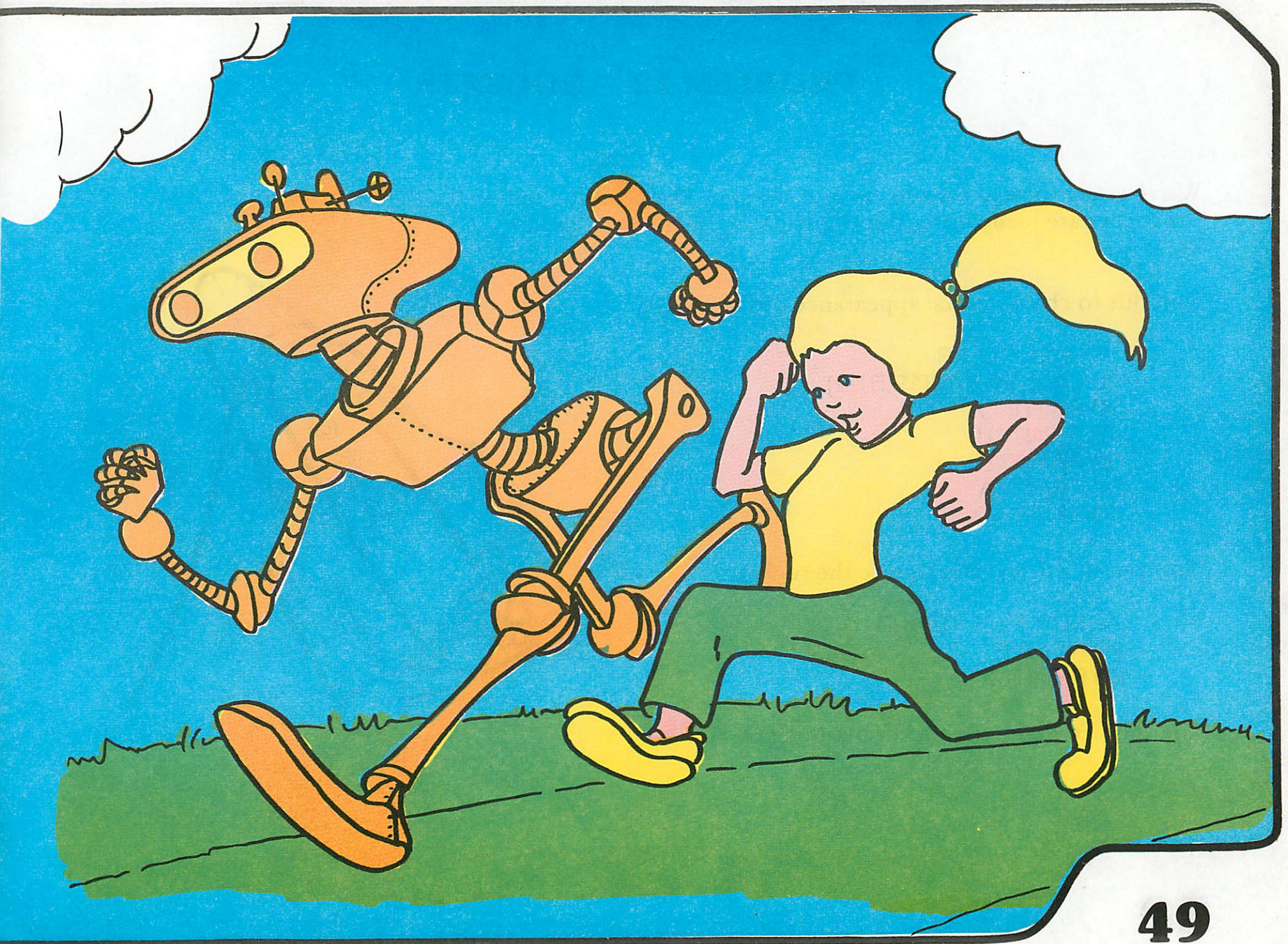
To see how to do it, enter and run this program:

```
10 CALL CLEAR
20 PRINT "ONE MINUTE, PLEASE"
30 FOR T = 1 TO 18000
40 NEXT T
50 PRINT
60 PRINT "NOW WHAT?"
```

Get a watch or stopwatch and time how long it takes from the time you press RETURN after typing RUN until the NOW WHAT? appears. Then change the 18000 in line 20 to each of the following numbers and time the delay.

18000	_____	sec.
36000	_____	sec.
72000	_____	sec.
144000	_____	sec.





PROBLEM 12 ROCKETS

We're going to send rockets up into the sky. The nose and tail of the rocket are made using the \wedge (either on the N or on the 6 key).

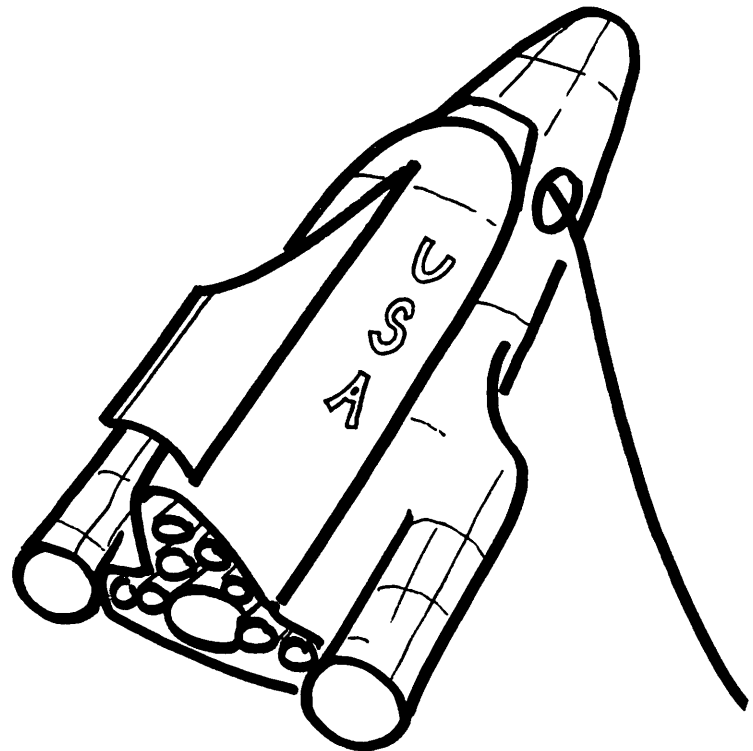
Run this to check on the appearance of the rocket.

```
10 CALL CLEAR
20 CALL VCHAR(19,5,32)
30 PRINT " ^ "
40 PRINT "U"
50 PRINT "S"
60 PRINT "A"
70 PRINT " ^ "
```

Now put in a delay; then ignite the rocket.

```
75 FOR T = 1 TO 400
76 NEXT T
80 PRINT "X"
```

Run this. To make the rocket blast off, GOTO 20.



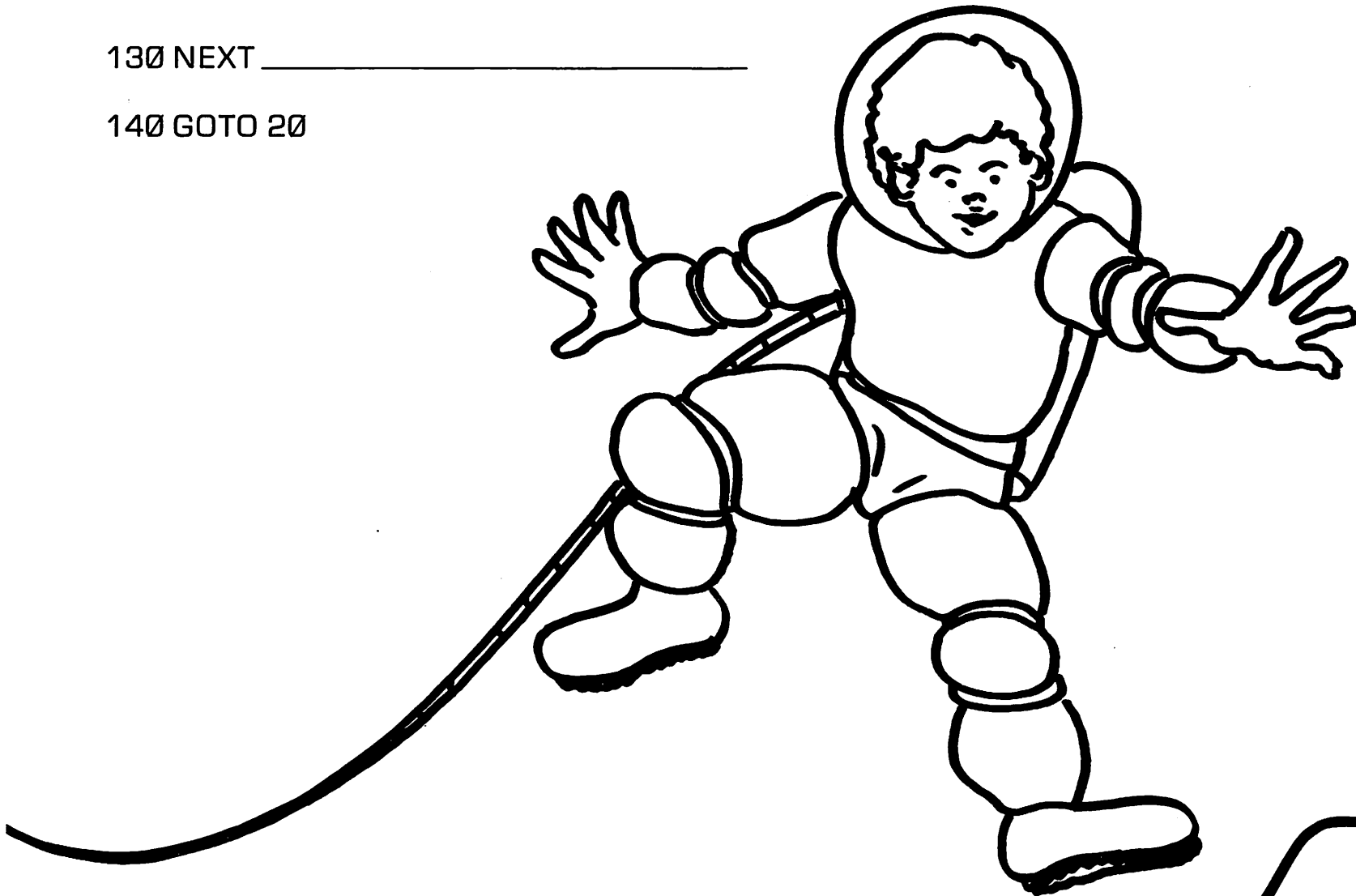
Put in a delay first, so you can watch the ignition stage.

120 FOR _____

125 PRINT

130 NEXT _____

140 GOTO 20



PROBLEM 13 COUNTING

To watch the computer count, enter this program and run it:

```
10 FOR K = 1 TO 15  
20 PRINT K  
30 NEXT K
```

Change line 10 so the numbers from 1 to 100 appear:

10 _____

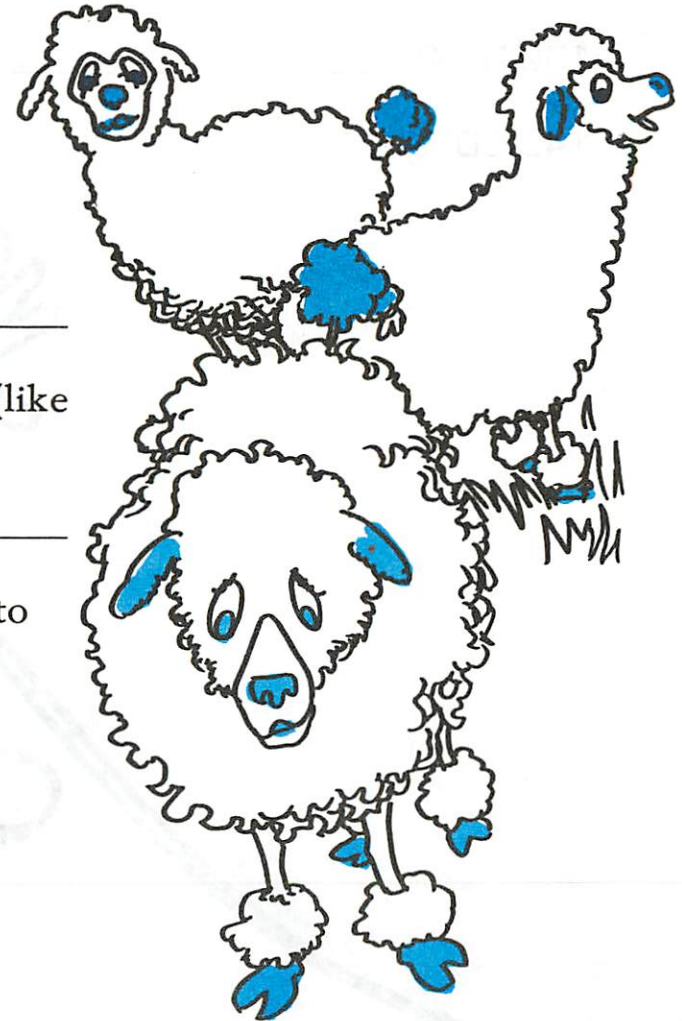
Change line 20 so all 100 numbers are on the screen at one time (like Wallpaper):

20 _____

Get out your watch and time how long it takes the computer to count to 1000 if it has to take time to write the numbers.

It takes

_____ sec.

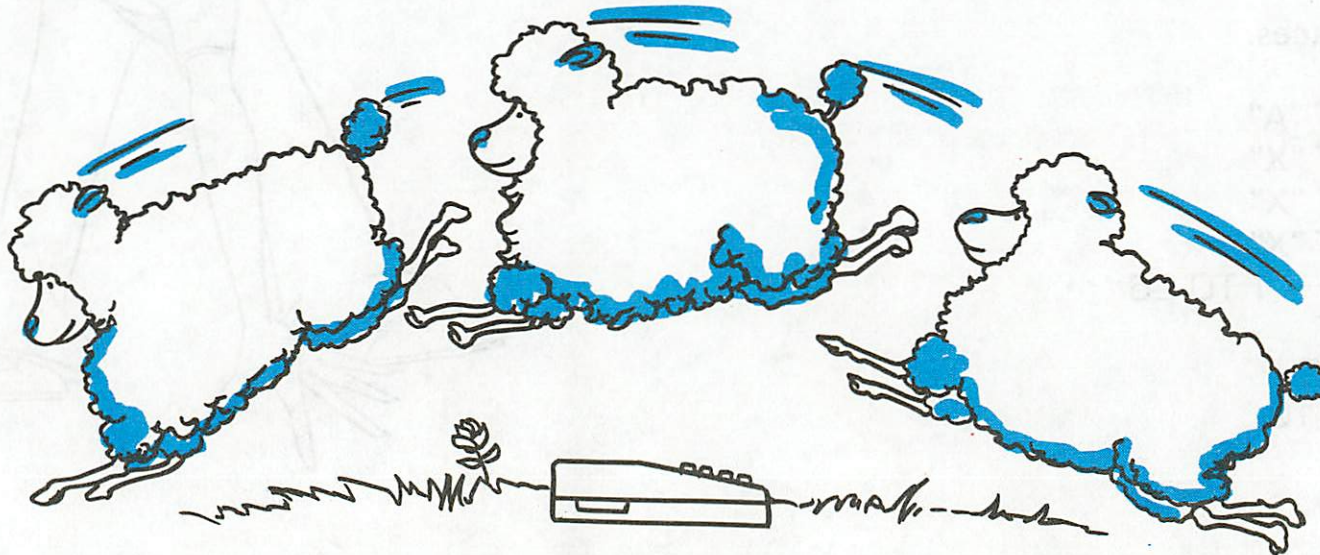


Run the next program three times to test your reflexes.

```
10 CALL CLEAR
20 PRINT "WHEN THE NUMBERS APPEAR"
30 PRINT "PRESS FCTN 4"
40 FOR T = 1 TO 1500
50 NEXT T
60 FOR K = 1 TO 500
70 PRINT K
80 NEXT K
```

What were your reaction numbers? _____

How can you tell if someone presses FCTN 4 before the numbers appear?



PROBLEM 14 TOUCH YOUR TOES

The FOR . . . NEXT loop (that's what it's called whenever you use FOR and NEXT to count something) can be used to do something a fixed number of times. Try each of these to see how:

A. It can print a message.

```
10 FOR M = 1 TO 10  
20 PRINT "TOO BAD"  
30 NEXT M
```

Change line 20 to print something else.

B. It can skip spaces.

```
10 PRINT "A"  
20 PRINT "X"  
30 PRINT "X"  
40 PRINT "X"  
50 FOR S = 1 TO 25  
60 PRINT  
70 NEXT S  
80 GOTO 10
```

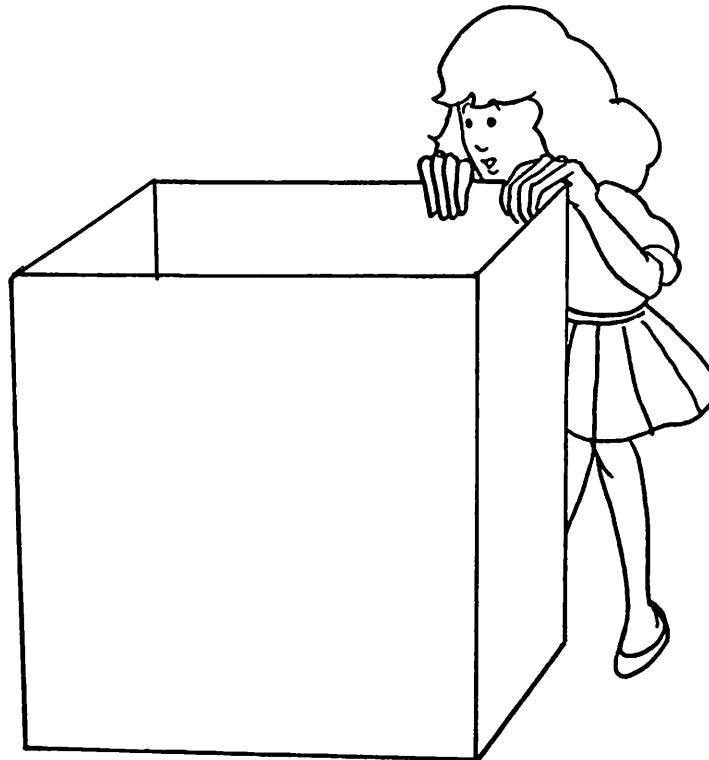


C. It can draw a box.

```
10 PRINT "#####"  
20 FOR S = 1 TO 10  
30 PRINT "# ... #"  
40 NEXT S  
50 PRINT "#####"
```

Erase the dots in line 30 so the box is empty.

Write a program which draws a high, wide box.



PROBLEM 15 MOVERS

Enter and run this program:

```
100 CALL CLEAR 110 FOR I = 1 TO 32
120 CALL HCHAR(12,I,42)
160 NEXT I
```

What happens? _____

Slow it down a little by adding these lines:

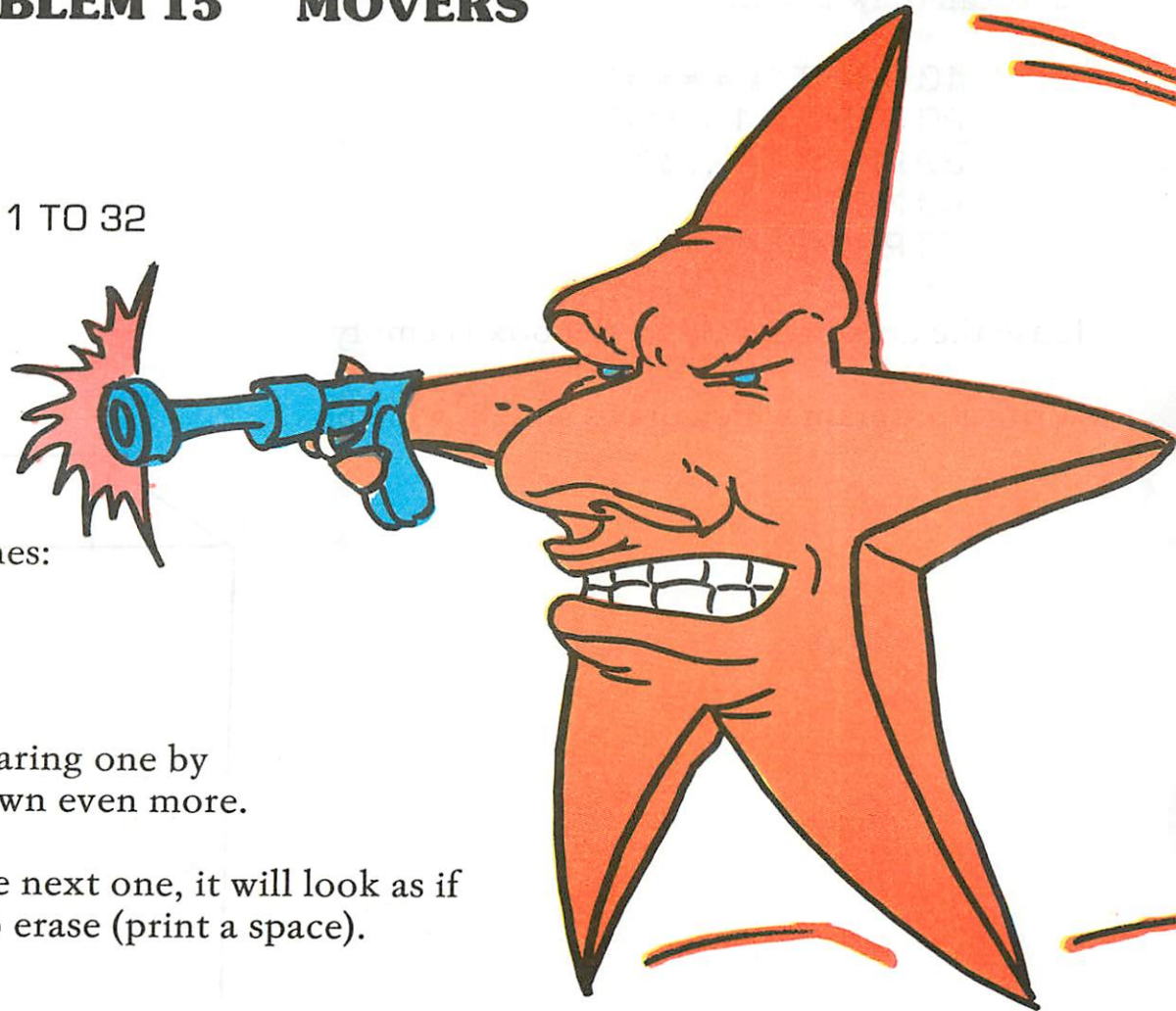
```
130 FOR DELAY = 1 TO 50
140 NEXT DELAY
```

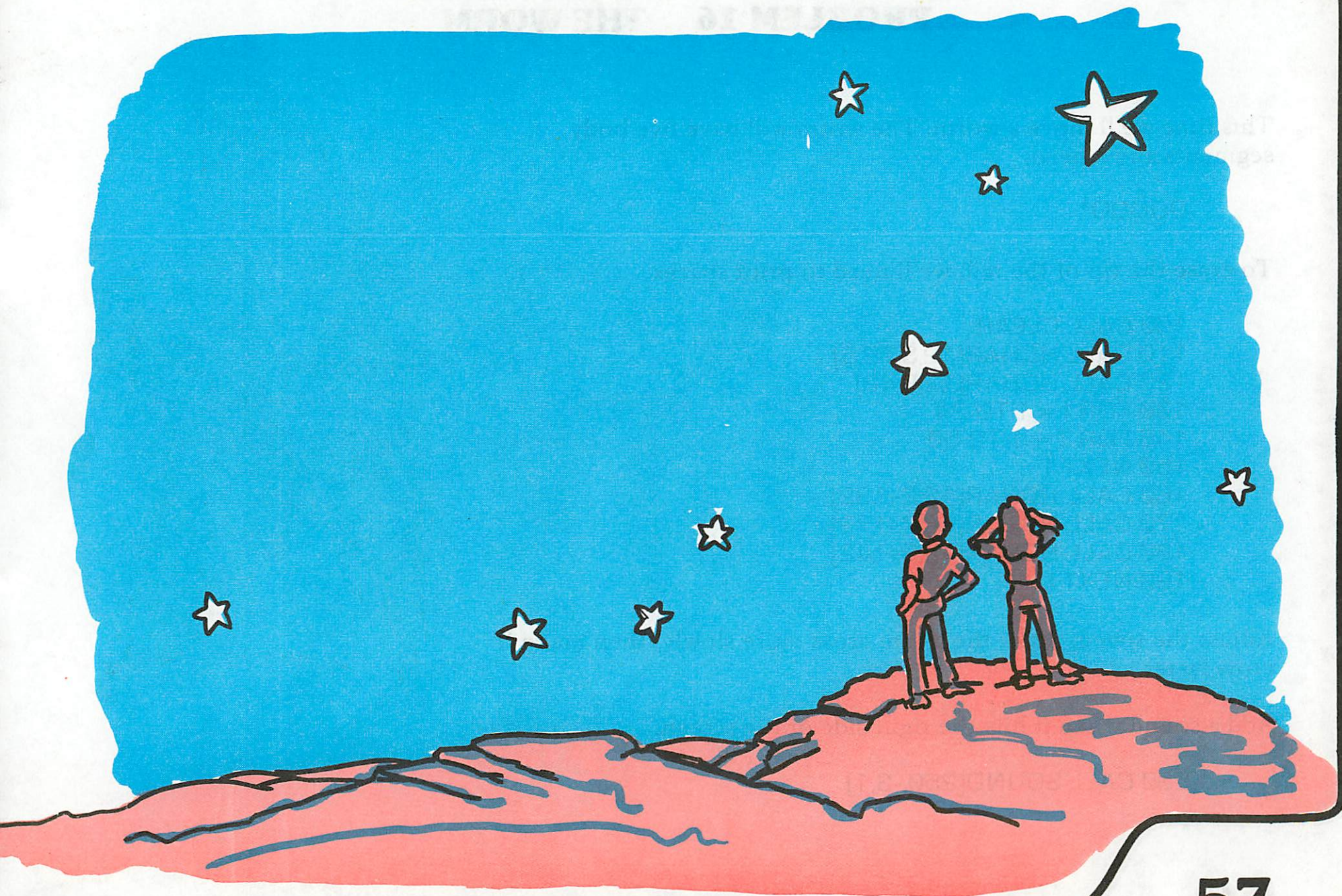
Run it now. Can you see the stars appearing one by one? _____ If not, slow it down even more.

If we erase each star before printing the next one, it will look as if just one star is moving. Add line 150 to erase (print a space).

```
150 CALL HCHAR(12,I,32)
```

Run it now. Does the last * stay or disappear? _____





PROBLEM 16 THE WORM

This time we'll move a worm. The worm will have five body segments and a head.

```
00000*
```

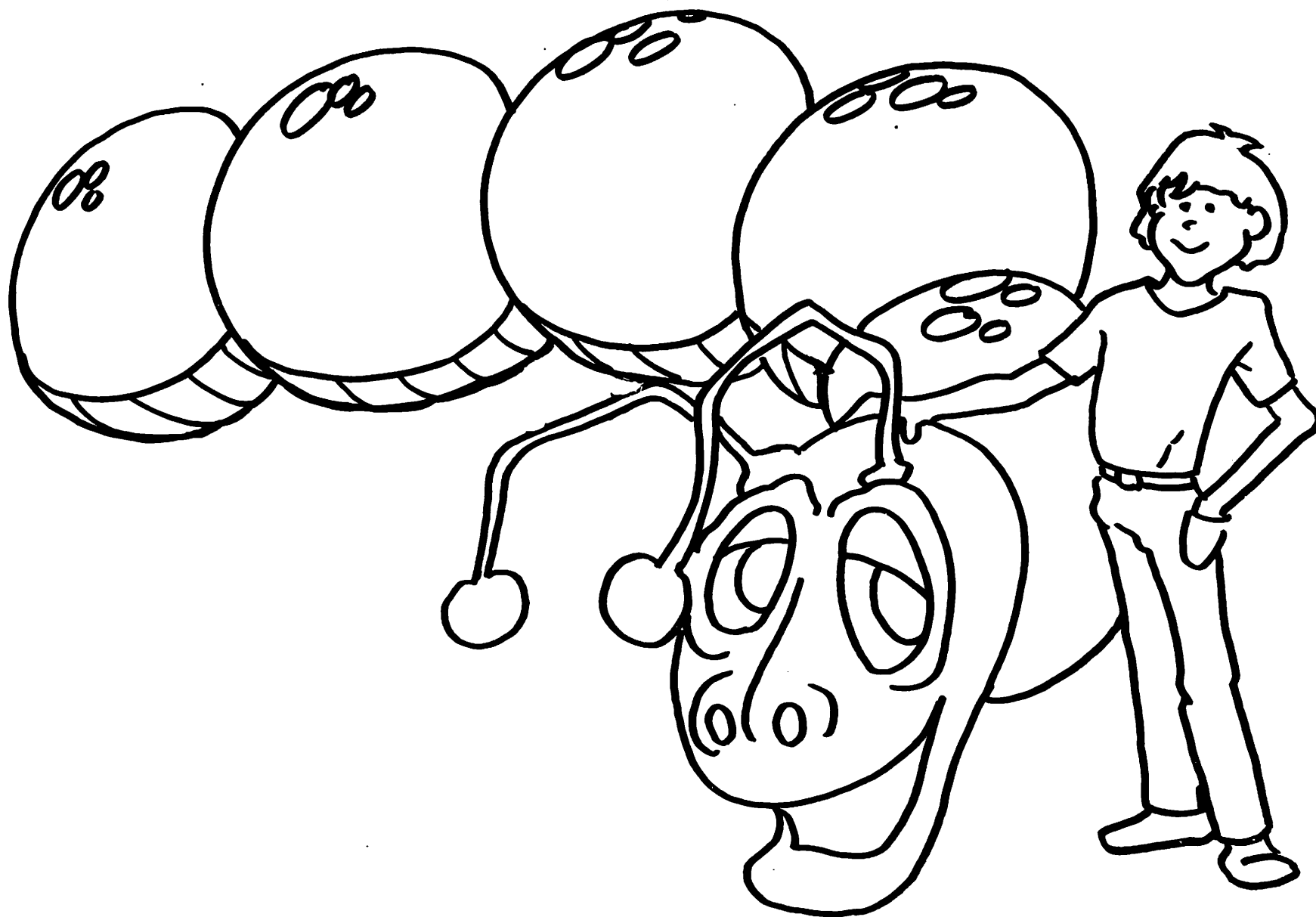
To erase the tip of the tail, we'll need to print spaces.

```
100 CALL CLEAR
110 CALL HCHAR(12,3,79,5)
120 CALL HCHAR(12,8,42)
130 FOR I = 1 TO 24
140 FOR T = 1 TO 50
150 NEXT T
160 CALL HCHAR(12,2+I,32)
170 CALL HCHAR(12,7+I,79)
180 CALL HCHAR(12,8+I,42)
190 NEXT I
```

Adjust the speed; make the worm cross more slowly, then make it cross faster.

Add a sound when it hits the right side of the screen.

```
200 CALL SOUND(250,-3,1)
```



PROBLEM 17 MORE COUNTING

This program lets the user tell it how high to count. Because the user is telling it a NUMBER, not a word, the \$ is not used.

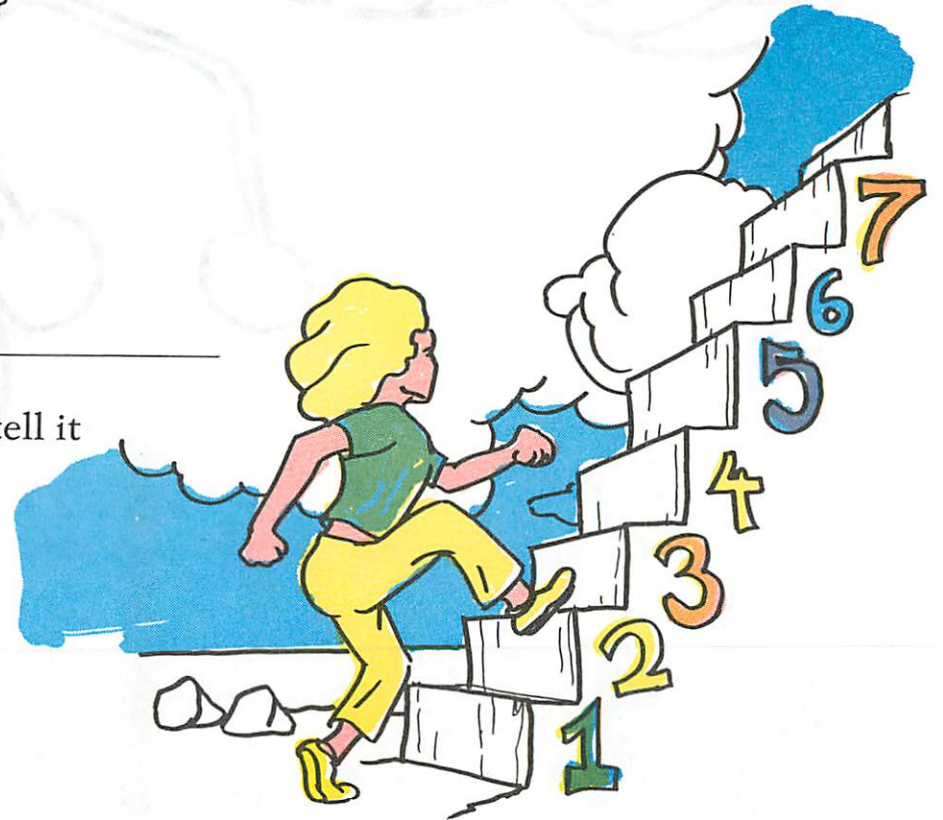
How High will be called HH in the program.

Enter and run this program:

```
10 PRINT "HOW HIGH SHOULD I COUNT?"
20 INPUT HH
30 FOR K = 1 TO HH
40 PRINT K
50 NEXT K
```

What happens if you tell it TEN, instead of 10?

(When the computer expects a number, you must tell it a number.)

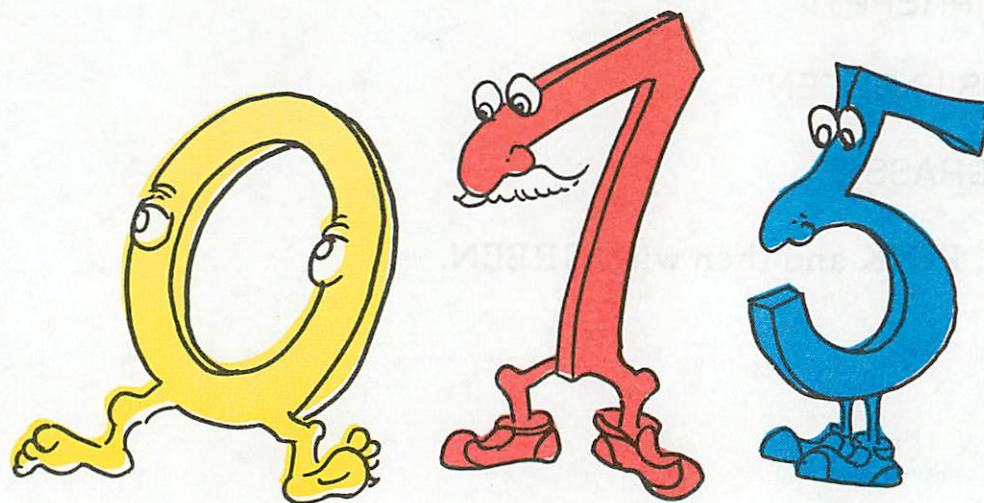


Add a line to make the program print

THE END

when it has finished counting.

What punctuation mark at the end of line 40 will cause the numbers to be printed across the screen?



PROBLEM 18 TESTING WITH IF ... THEN

In BASIC two words, or two numbers, can be tested to see whether they are the same or different.

The symbol for "is the same as" is the equal sign, "=".

The symbol for "is not the same as" is "< >".

Enter and run this program:

```
10 CALL CLEAR
20 INPUT "TELL ME A COLOR ":C$
30 IF (C$<>"GREEN") & (C$<>"PINK") THEN 80
40 IF C$ = "GREEN" THEN 60
50 PRINT "VERY PREPPY!"
55 END
60 PRINT "GRASS IS GREEN"
70 END
80 PRINT "NOT GRASSY!"
```

Try answering with RED, PINK and then with GREEN.





PROBLEM 19 GOLDILOCKS

Complete and enter the Goldilocks program.

```
10 CALL CLEAR  
20 PRINT "WHOSE PORRIDGE IS IT?"  
30 PRINT "TASTE IT AND ENTER HOT, COLD OR OK"  
40 INPUT A$  
50 IF A$ <> "OK" THEN 60 ELSE 70
```

```
60 PRINT _____  
65 END
```

```
70 PRINT _____  
80 END
```





PROBLEM 20 ANIMAL GUESS

Now we'll use IF . . . THEN to play a guessing game. First we'll set the animal to be a HORSE, then we'll add extra lines so the mystery animal can be typed in.

Look at the program. What happens after line 140 if the guess was HORSE?

```
100 AN$ = "HORSE"  
110 CALL CLEAR  
120 PRINT "I'M THINKING OF AN ANIMAL"  
130 PRINT "GUESS"  
140 INPUT G$  
150 IF G$ < > AN$ THEN 160 ELSE 180  
160 PRINT "NO"  
170 GOTO 130  
180 PRINT
```

Once the program is working, add these lines:

```
10 PRINT "THINKER, TELL ME AN ANIMAL"  
100 INPUT AN$  
115 PRINT "GUESSER'S TURN"
```

The thinker types in an animal while the guesser isn't looking.
Then the guesser tries to guess it.



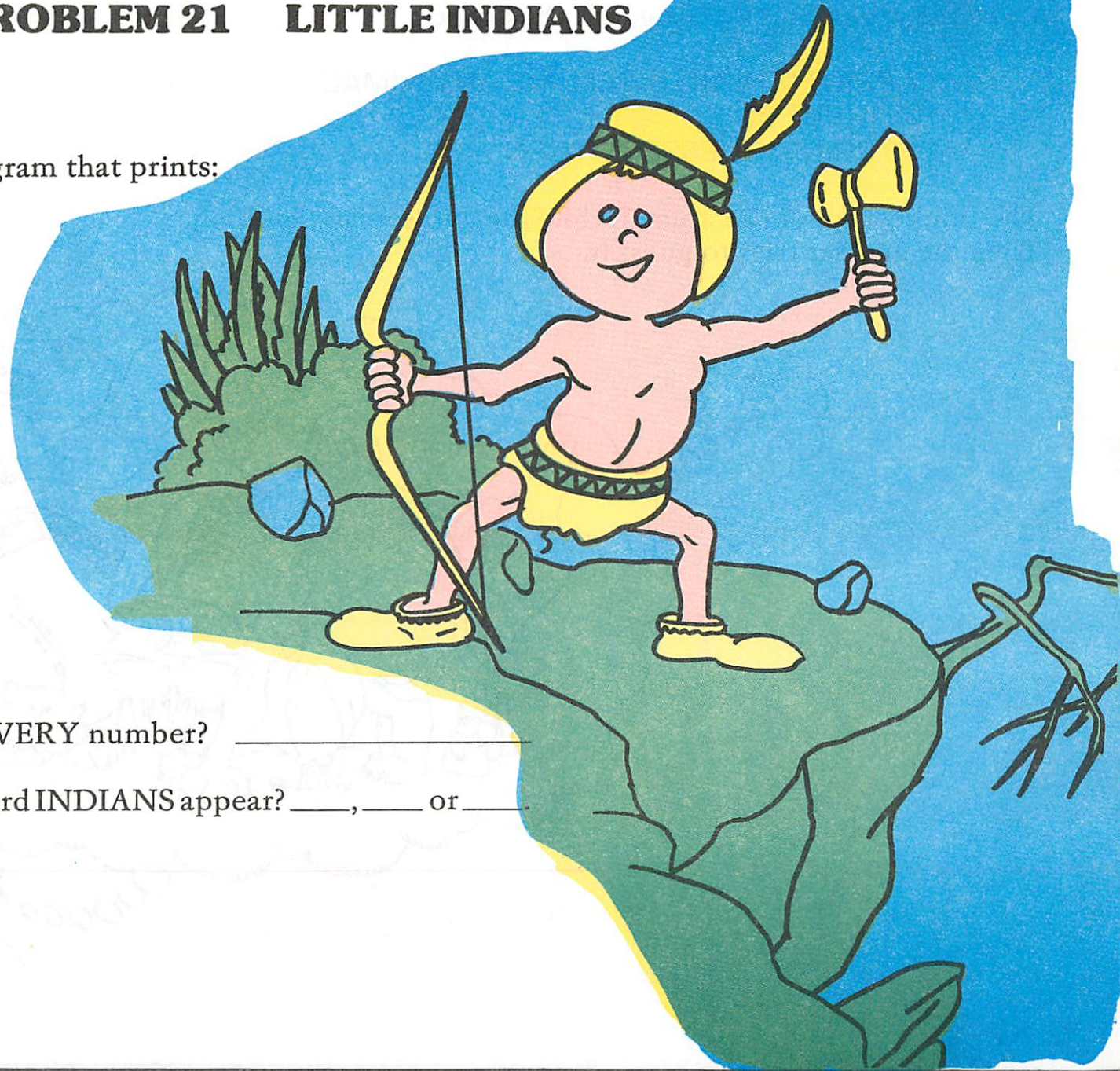
PROBLEM 21 LITTLE INDIANS

We're going to write a program that prints:

1 LITTLE
2 LITTLE
3 LITTLE
INDIANS
4 LITTLE
5 LITTLE
6 LITTLE
INDIANS
7 LITTLE
8 LITTLE
9 LITTLE
INDIANS
10 LITTLE
NAVAJO KIDS

What word appears after EVERY number? _____

What numbers make the word INDIANS appear? _____, _____ or _____



Complete the program and run it:

```
10 FOR K = 1 TO 3
20 PRINT K; " LITTLE"
30 NEXT K
35 PRINT "INDIANS"
40 FOR K = 4 TO 6
45 PRINT K; " LITTLE"
50 NEXT K
55 PRINT _____

60 FOR K = _____
65 PRINT _____
70 NEXT _____

75 PRINT _____

80 PRINT "10 LITTLE "
85 PRINT "NAVAJO KIDS"
90 END
```

Change lines 35, 55 and 75 to

GOSUB 95

and add these lines. See what happens.

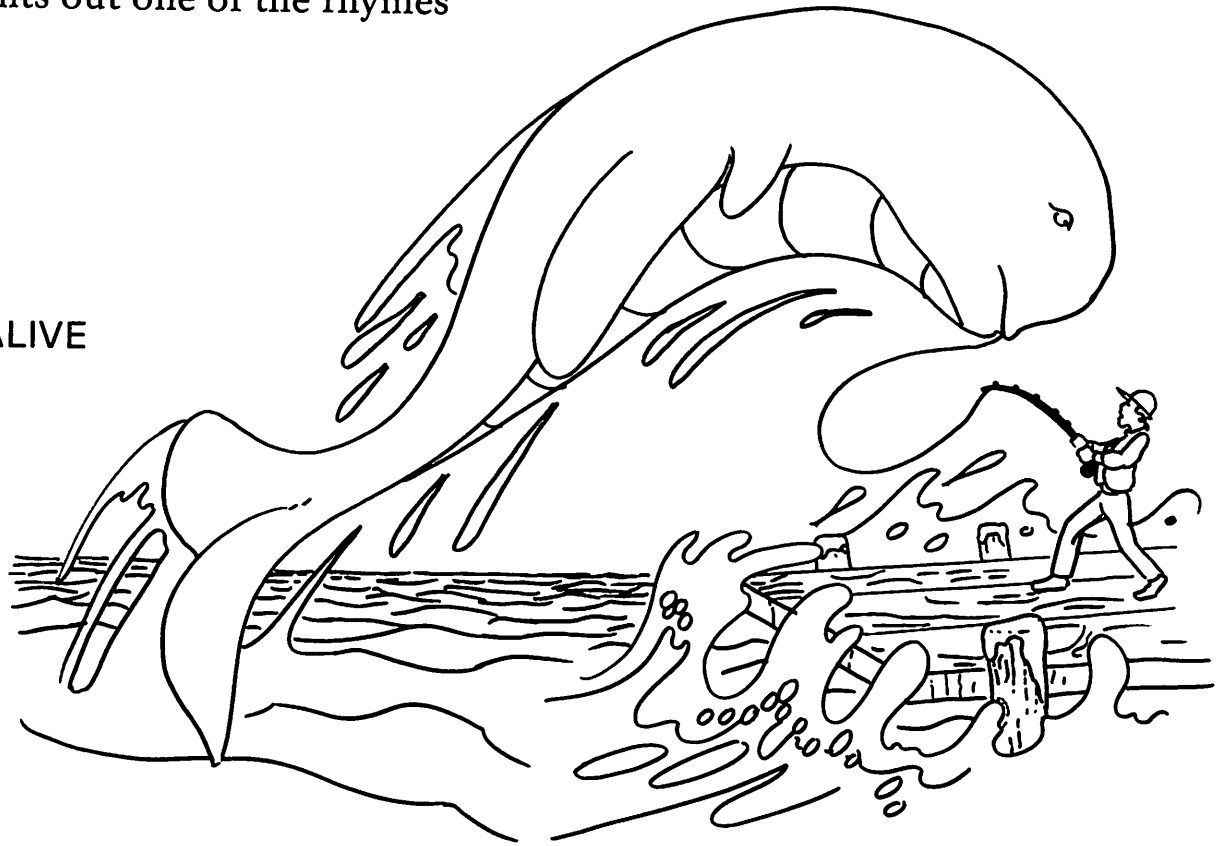
```
95 PRINT "INDIANS"
100 RETURN
```



PROBLEM 22 NURSERY RHYMES

Now you write a program that prints out one of the rhymes below.

1
2
3
4
5
6 ONCE I CAUGHT A FISH ALIVE
7
8
9
10
11 THEN I LET IT GO AGAIN



or this one

```
1
2
3
4
MARY AT THE COTTAGE DOOR
5
6
7
8
EATING CHERRIES OFF A PLATE
```

Use this space to record your program:

10 FOR _____

20 PRINT _____

30 NEXT _____

40 PRINT _____

50 FOR _____

60 PRINT _____

70 NEXT _____

80 PRINT _____



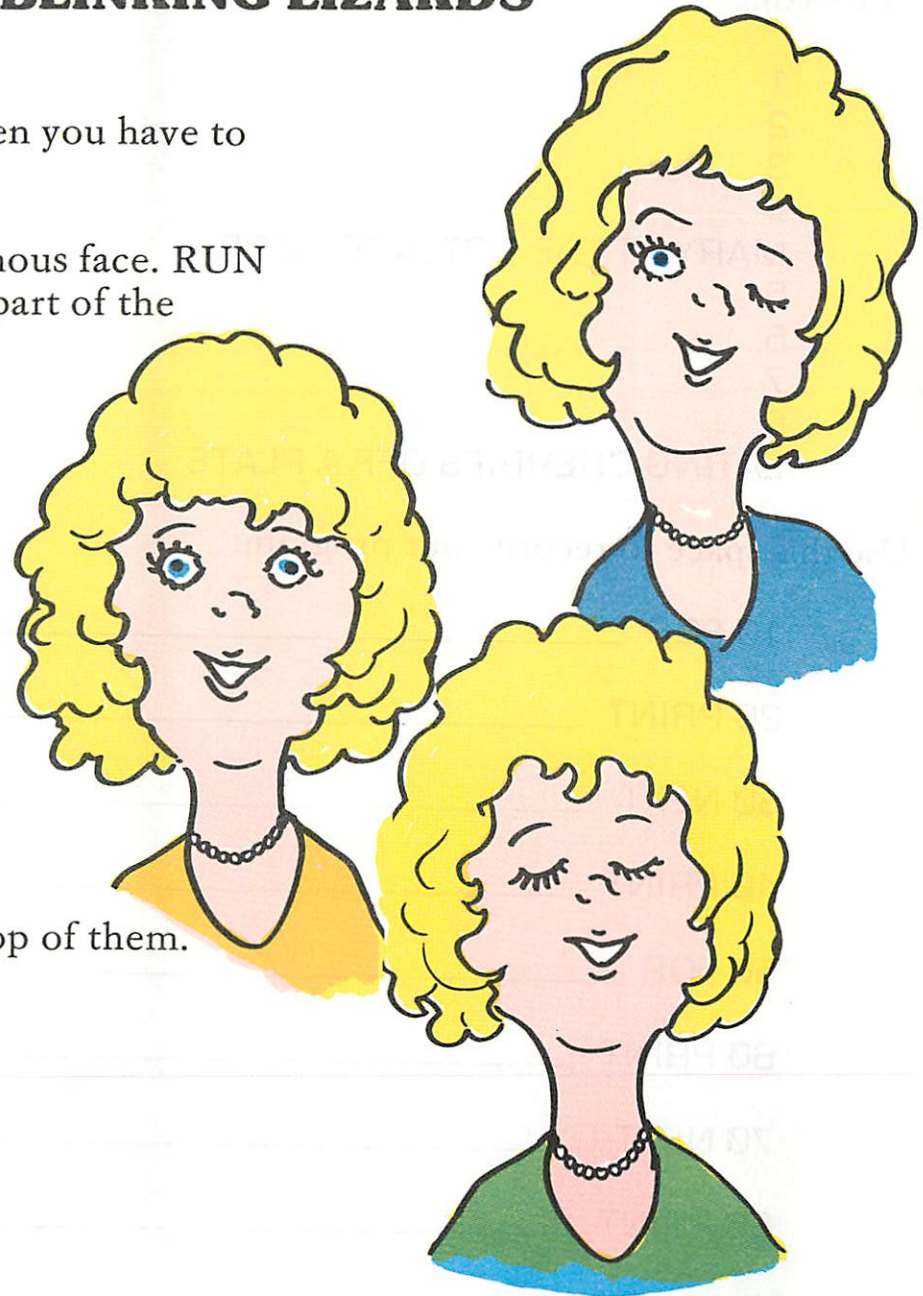
PROBLEM 23 BLINKING LIZARDS

When you want only part of a picture to move, then you have to print on top of the picture.

Type in and run this program to draw part of a famous face. RUN the program after you enter each line to see what part of the picture it draws.

```
20 CALL CLEAR
50 CALL HCHAR(12,3,64,3)
60 CALL HCHAR(13,2,64,5)
70 CALL HCHAR(14,2,64)
71 CALL HCHAR(14,6,64)
80 CALL HCHAR(15,2,64)
81 CALL HCHAR(15,3,79)
82 CALL HCHAR(15,5,79)
83 CALL HCHAR(15,6,64)
90 FOR T = 1 TO 300
100 NEXT T
```

To change the eyes, the computer must print on top of them.



Add these lines:

```
110 CALL HCHAR(15,2,64)
111 CALL HCHAR(15,3,95)
112 CALL HCHAR(15,5,95)
113 CALL HCHAR(15,6,64)
```

Run the program now.

To keep the eyes blinking, first put in a pause while they're closed:

```
120 FOR DELAY = 1 TO 300
130 NEXT DELAY
```

To keep the eyes blinking, add a GOTO.

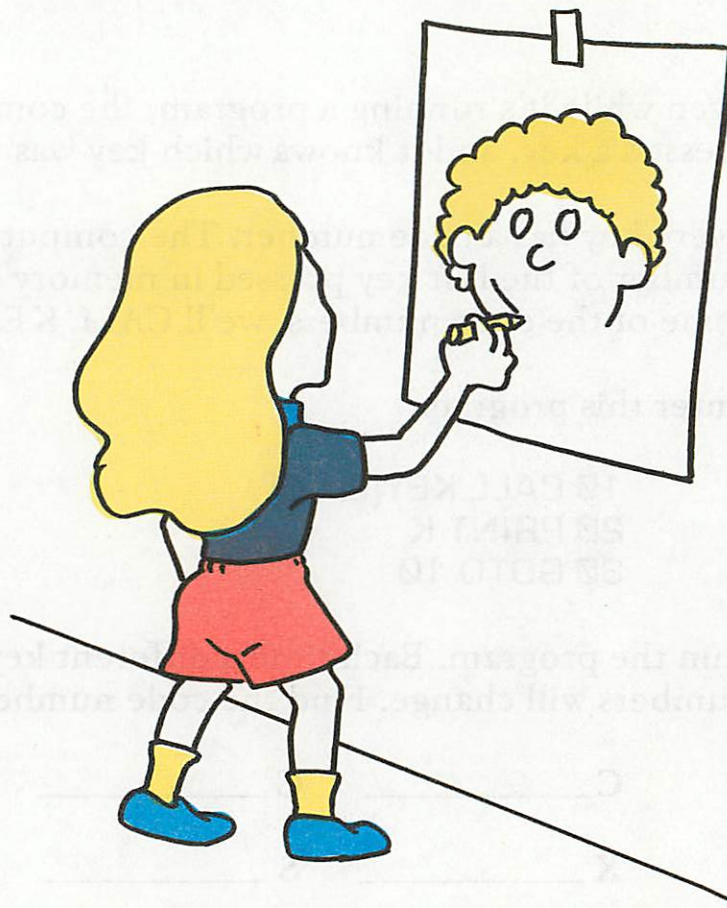
```
140 GOTO 80
```

To give the character red hair and blue eyes against a white background, add these lines:

```
10 CALL SCREEN(16)
30 CALL COLOR(5,7,16)
40 CALL COLOR(6,5,16)
```

You'll have to use FCTN 4 to stop this.

See if you can make her wink instead of blink.



PROBLEM 24 PEEKABOO - THE COMPUTER IS WATCHING YOU

Even while it's running a program, the computer knows if you've pressed a key, and it knows which key was the last one touched.

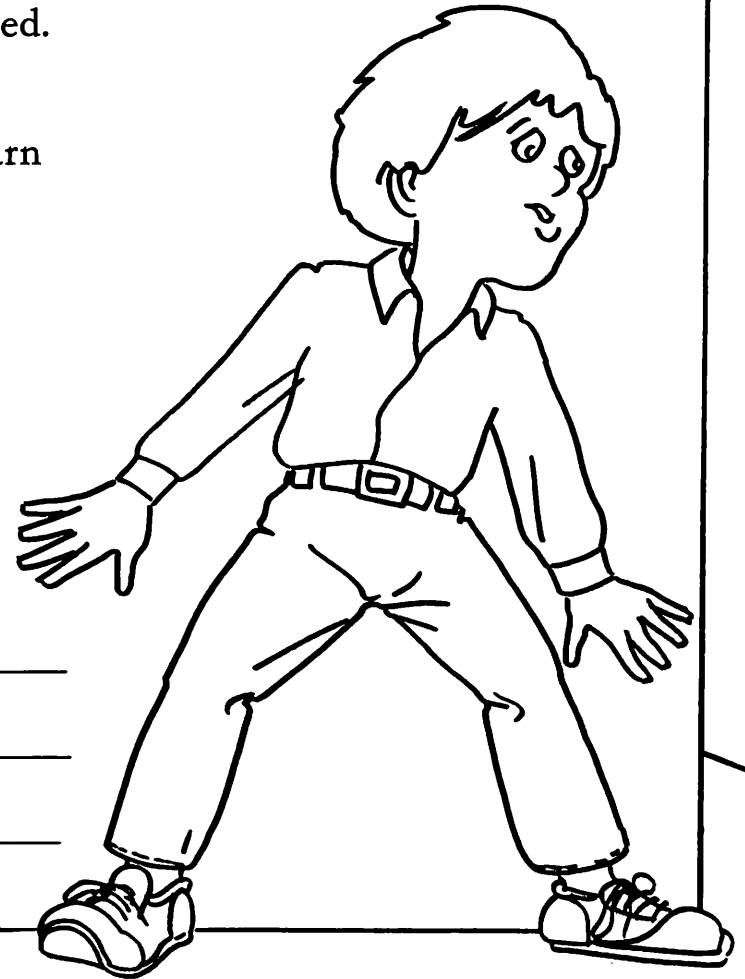
Every key has a code number. The computer keeps the code number of the last key pressed in memory location K. To learn some of the code numbers, we'll CALL KEY location K.

Enter this program:

```
10 CALL KEY(0,K,S)  
20 PRINT K  
30 GOTO 10
```

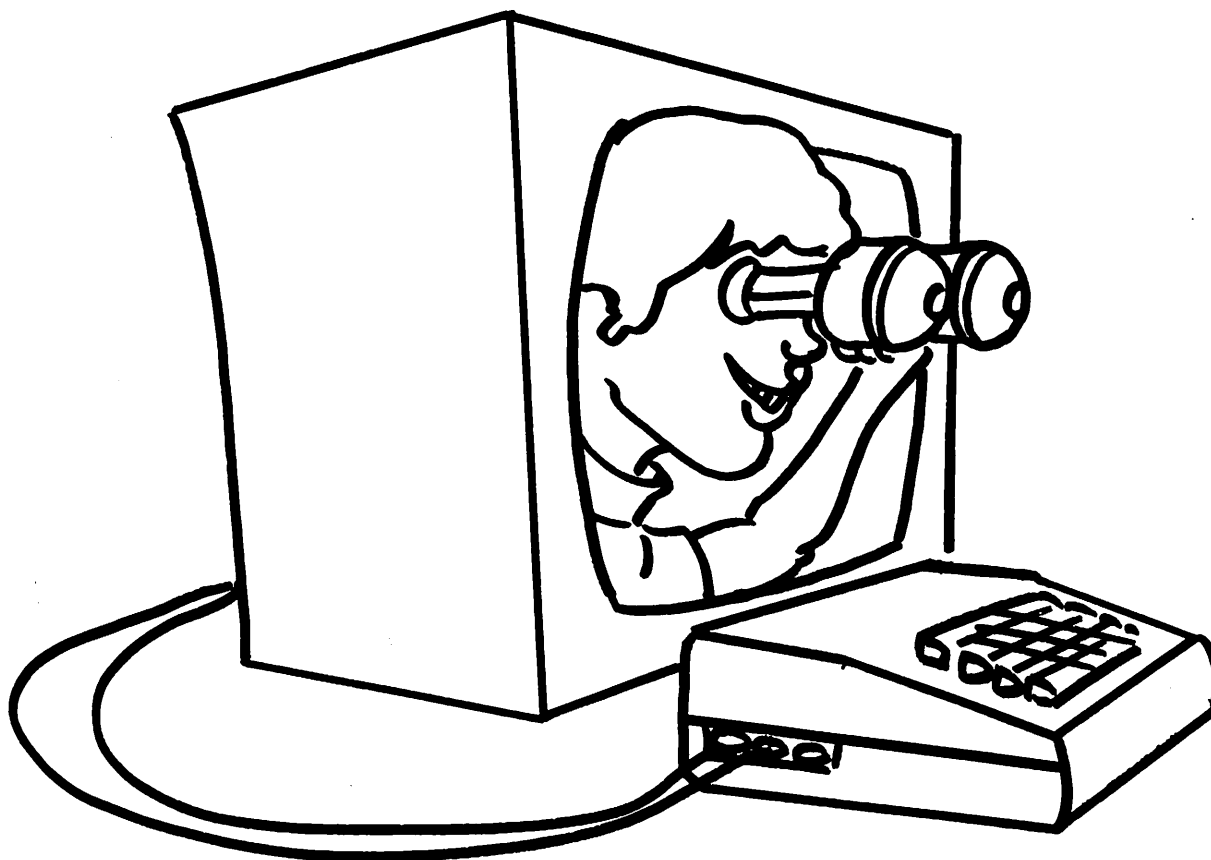
Run the program. Each time a different key is pressed, the numbers will change. Find the code numbers for these keys:

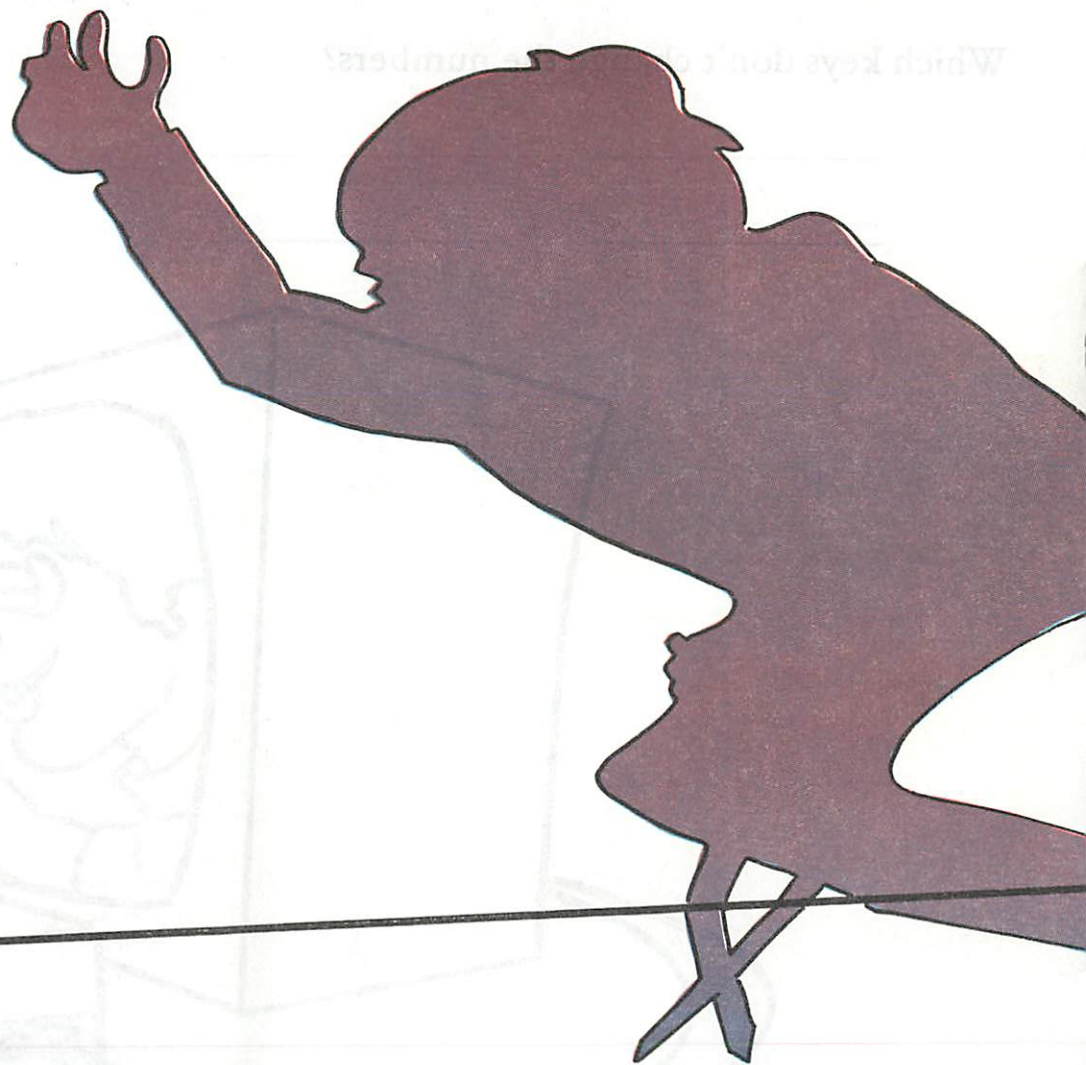
C _____	G _____	space bar _____
X _____	S _____	RETURN _____
Z _____	4 _____	* _____

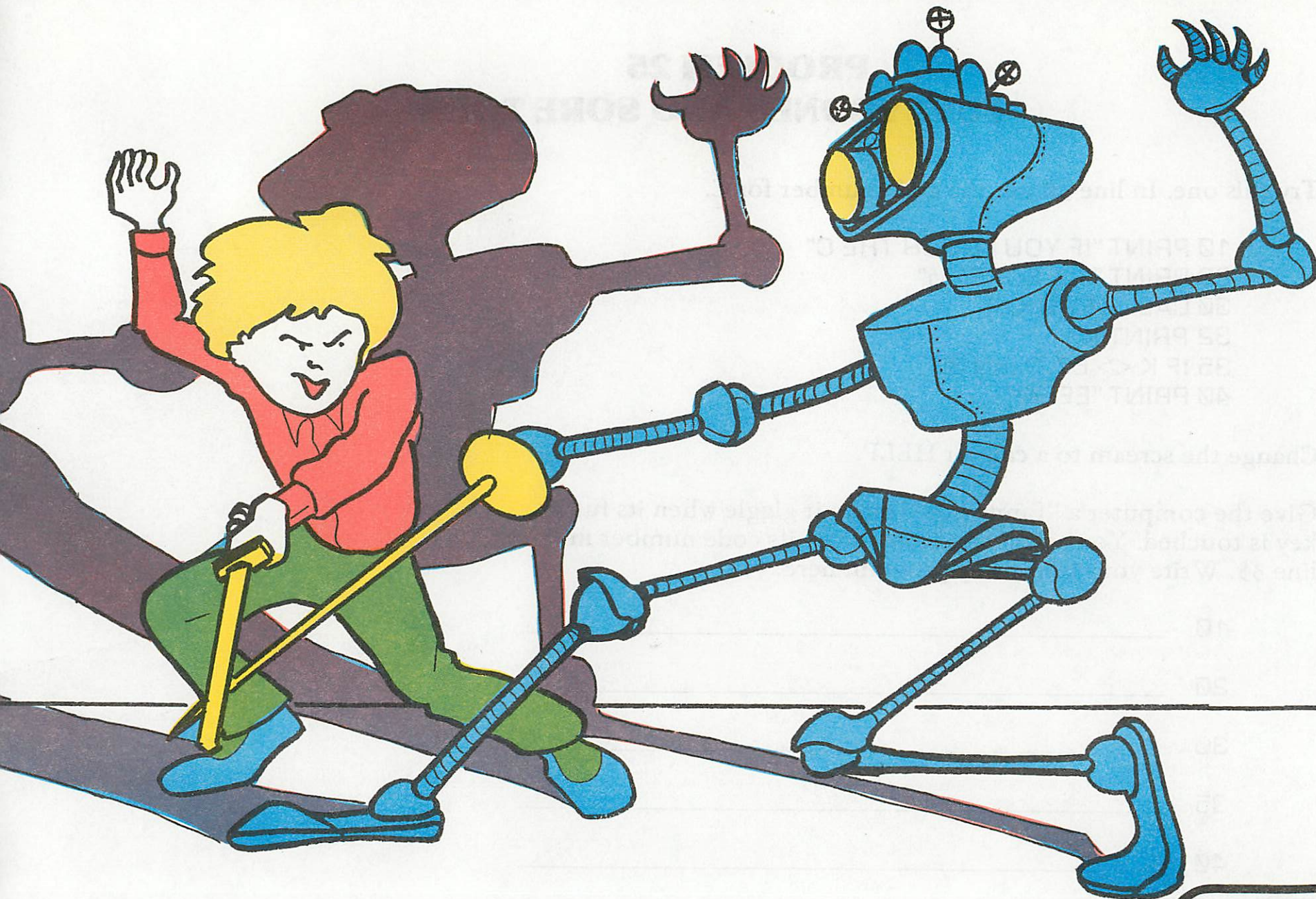


What happens when you touch FCTN 4? _____

Which keys don't change the numbers?







PROBLEM 25

FUNNYBONES AND SORE TOES

Try this one. In line 30 use the code number for C.

```
10 PRINT "IF YOU TOUCH THE C"  
20 PRINT "I'LL SCREAM"  
30 CALL KEY(0,K,S)  
32 PRINT K  
35 IF K <> 67 THEN 30  
40 PRINT "EEEK!"
```

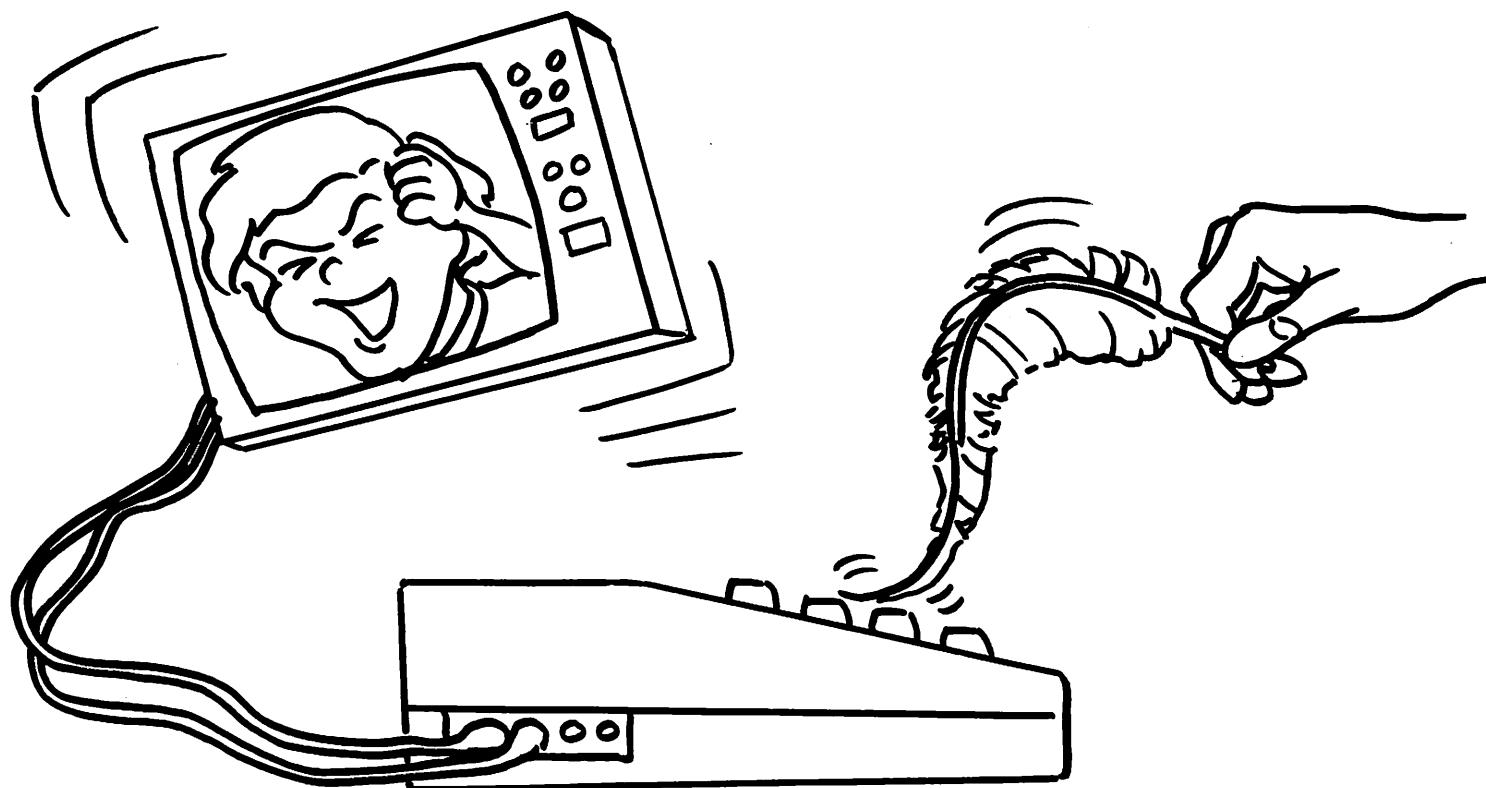
Change the scream to a call for HELP.

Give the computer a "funny key"; make it giggle when its funny key is touched. You select the key and use its code number in line 35. Write your funny-key program here.

```
10 _____  
20 _____  
30 _____  
35 _____  
40 _____
```

Have the computer make a sound when its secret key is touched.
Add this line:

```
45 CALL SOUND(250,1000,2)
```



PROBLEM 26 QUIZ TIME

Read the program. How many questions are on the quiz?_____

Complete line 110 so that the color question is asked again, if the answer is not GREEN.

Then make up the last question on the quiz; use lines 130 - 170.

```
10 CALL CLEAR
20 PRINT "QUIZ TIME"
30 PRINT
40 PRINT "WHAT IS THE CAPITAL OF MICHIGAN?"
50 INPUT A$
60 IF A$ <> "LANSING" THEN 70 ELSE 80
70 PRINT "NO, LANSING"
75 GOTO 90
80 PRINT "RIGHT"
90 PRINT "WHAT DO BLUE AND YELLOW MAKE?"
95 INPUT A$
100 IF A$ <> "GREEN" THEN 105 ELSE 120
105 PRINT "TRY AGAIN"

110 GOTO _____
120 PRINT "GOOD"
```



130 PRINT _____

140 INPUT A\$

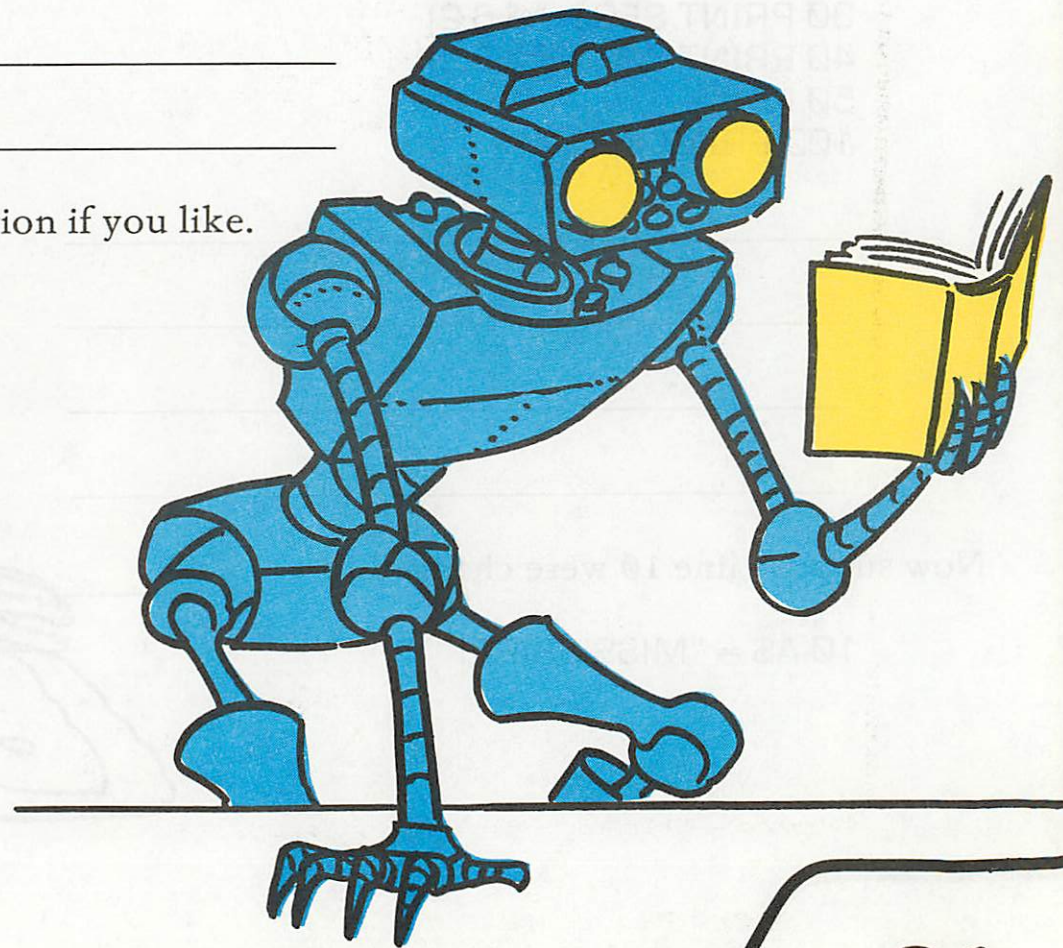
150 IF A\$ <> _____ THEN _____ ELSE _____

160 _____

170 _____

180 _____

Have a friend try your quiz. Add another question if you like.



PROBLEM 27 PIECES OF WORDS

Run this program. Copy the output, then try to answer the questions.

```
10 A$ = "BICYCLE"  
20 PRINT SEG$(A$,7,1)  
30 PRINT SEG$(A$,6,2)  
40 PRINT SEG$(A$,5,3)  
50 PRINT  
100 PRINT LEN(A$)
```

Now suppose line 10 were changed to

```
10 A$ = "MISSISSIPPI"
```



Predict what the output would be. Then check it.

Change lines 30, 40 and 50 so the output is

I
PI
PPI

See if you can make it print

M
MI
MISS

30

40

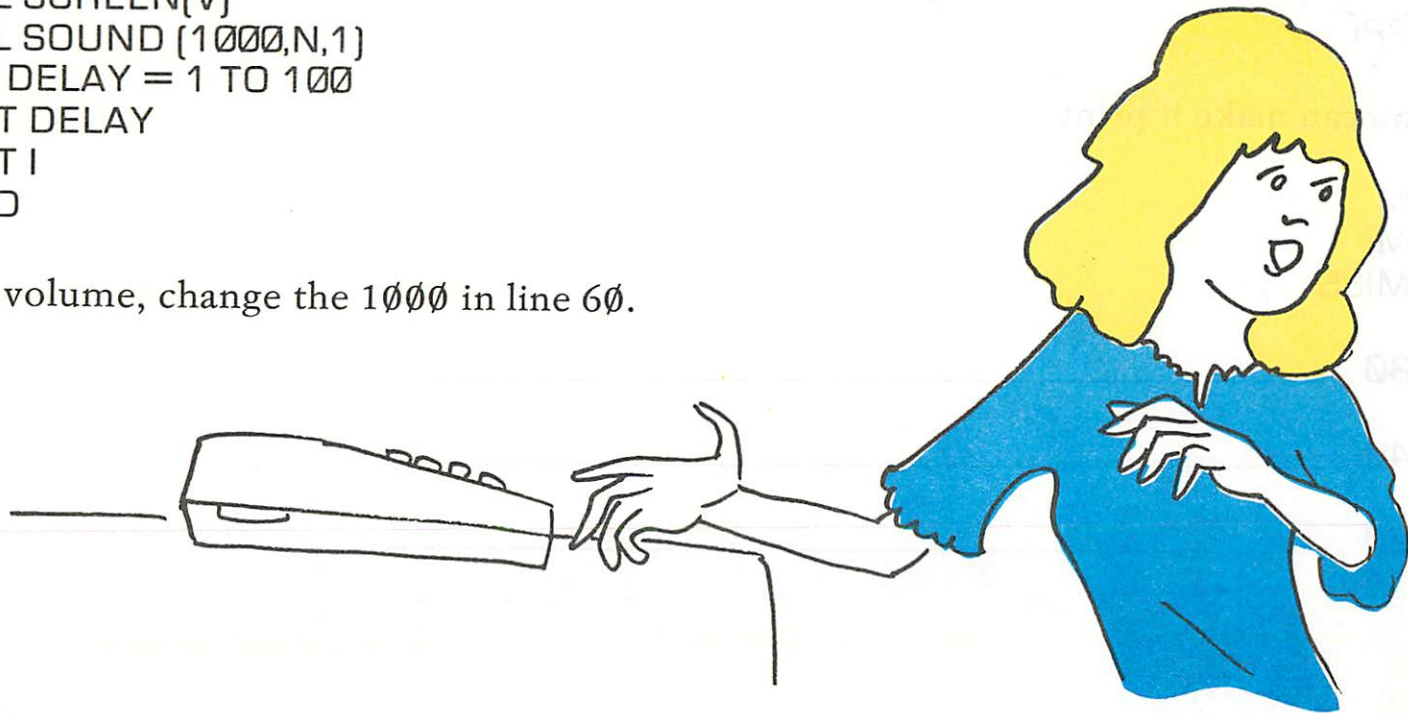
50

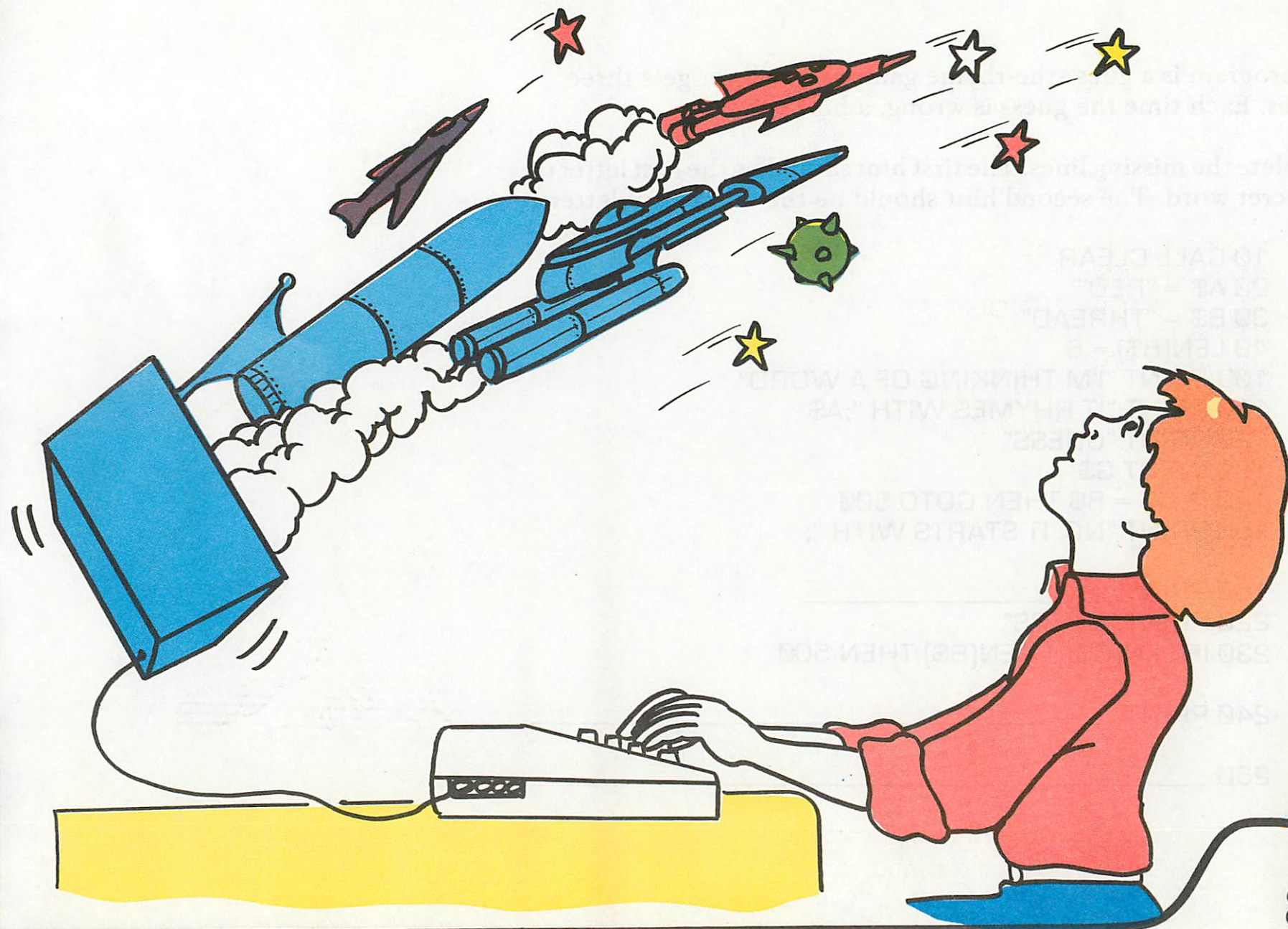
PROBLEM 28 COLOR AND SOUND

In this problem, the computer will “think” of a number from 1 to 255, play the note and color the screen.

```
10 CALL CLEAR
15 FOR I = 1 TO 50
20 RANDOMIZE
30 N = INT(395*RND)+131
40 V = INT(16*RND)+1
50 CALL SCREEN(V)
60 CALL SOUND (1000,N,1)
70 FOR DELAY = 1 TO 100
80 NEXT DELAY
90 NEXT I
100 END
```

To change the volume, change the 1000 in line 60.





PROBLEM 29 RHYMING GAME

This program is a guess-the-rhyme game. The player gets three guesses. Each time the guess is wrong, a hint is given.

Complete the missing lines. The first hint should be the first letter of the secret word. The second hint should be the number of letters.

```
10 CALL CLEAR
20 A$ = "RED"
30 B$ = "THREAD"
40 LEN(B$) = 6
100 PRINT "I'M THINKING OF A WORD"
110 PRINT "IT RHYMES WITH ";A$
120 PRINT "GUESS"
130 INPUT G$
140 IF G$ = B$ THEN GOTO 500
150 PRINT "NO. IT STARTS WITH ";
```

```
160 PRINT _____
220 PRINT "GUESS"
230 IF LEN(G$) = LEN(B$) THEN 500
```

```
240 PRINT _____
```

```
250 _____
```

320 PRINT "LAST GUESS"

330 _____

340 _____

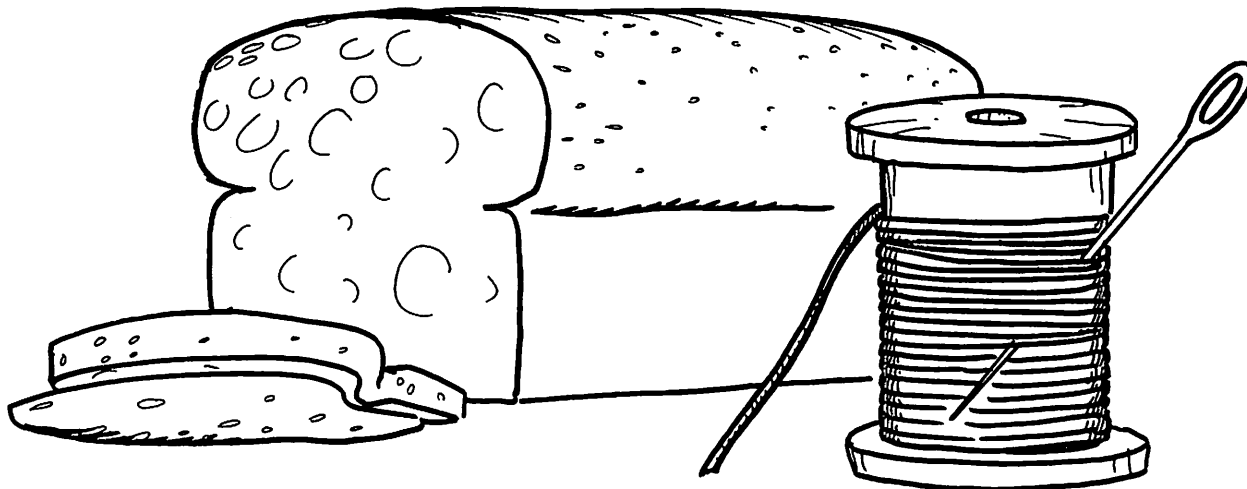
350 PRINT "NO, IT WAS ";

360 _____

370 END

500 PRINT _____

When the program is working, change A\$ and B\$ and let a friend try your game.



PROBLEM 30 COMPOUND WORDS

This program will make compound words.

```
10 PRINT "TELL ME TWO WORDS"  
20 PRINT "AND I'LL GLUE THEM TOGETHER"  
30 PRINT "FIRST WORD"  
40 INPUT A$  
50 PRINT "SECOND WORD"  
60 INPUT B$  
70 C$ = A$ & B$  
90 PRINT  
100 PRINT C$
```

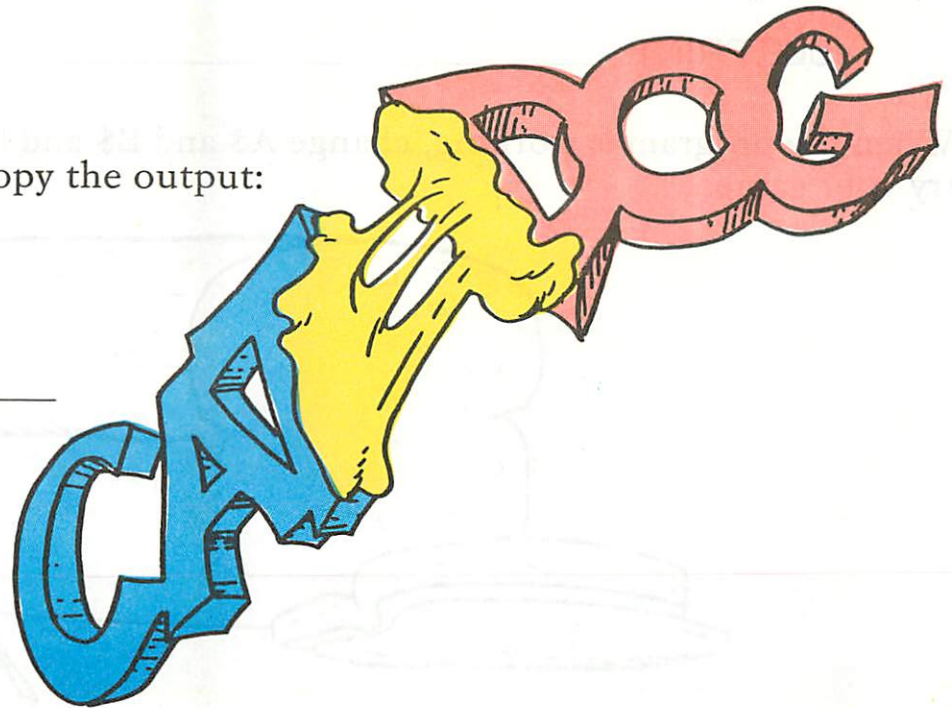
Run the program for these pairs of words and copy the output:

WORDS	OUTPUT
-------	--------

BOOK	
END	_____

UP	
SHOT	_____

UP	
CATCH	_____



Now change line 70 to

70 C\$ = B\$ & A\$

and run for the same sets of words. What happens?

Finally, add

75 C\$ = C\$ & C\$ & C\$

and run it for one of the pairs above. What happened?
(Remember, = means "is replaced by.")



PROBLEM 31 WORD PUZZLES

Look at this program. PREDICT the output. Then check it by entering and running the program.

```
10 A$ = "STAY"  
20 B$ = "RED"  
30 C$ = SEG$(A$,1,3) & SEG$(B$,1,1)  
40 D$ = SEG$(B$,1,1) & SEG$(A$,3,2) & SEG$(A$,1,1)  
50 PRINT C$,D$
```

Prediction _____

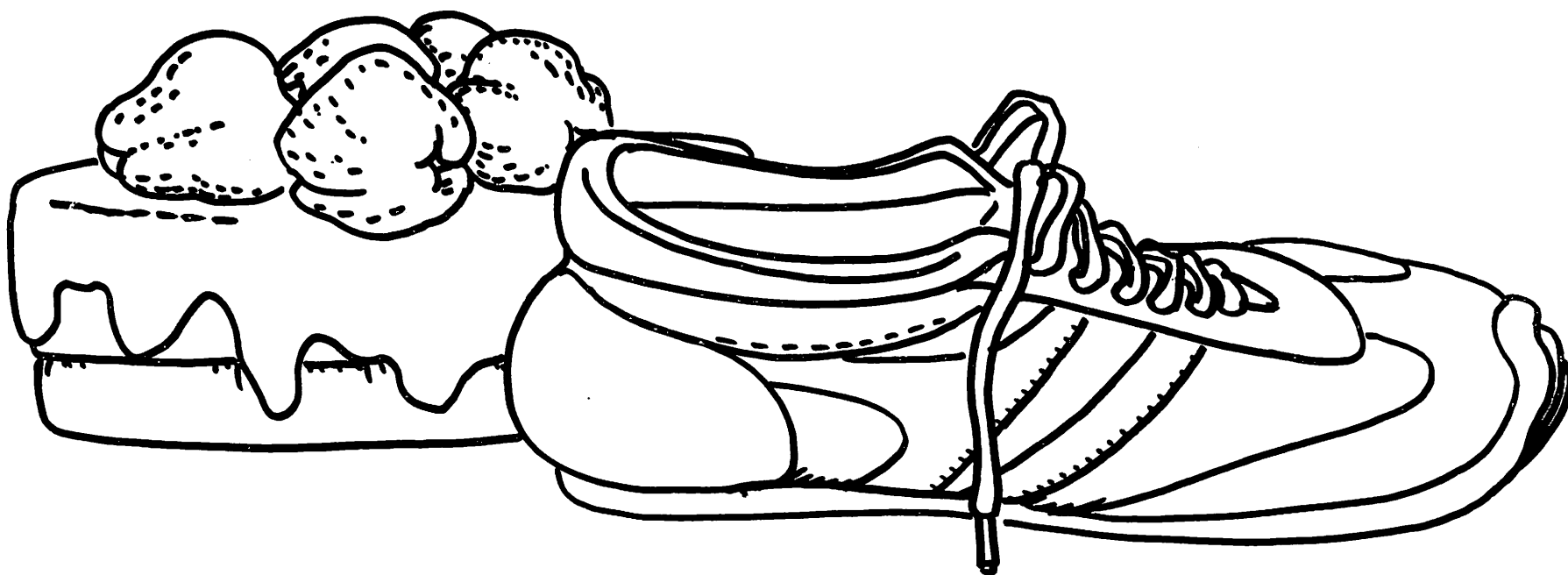
Now change A\$ to "STRAWBERRY".
Change B\$ to "SHORTCAKE".

Change lines 30 and 40 so that the output is

SHORTEST
STAKES

30 _____

40 _____



PROBLEM 32 DOING MATH

Enter and run this program:

```
10 PRINT "TELL ME A NUMBER"  
20 PRINT "AND I'LL DOUBLE IT"  
30 INPUT N  
40 PRINT 2*N
```

Check that the computer can double these numbers:

3 _____

20 _____

0 _____

-5 _____

Make the computer triple the number. Try it out on some numbers and write down the results.

(If you tried an enormous number, such as 1000000000000, your answer came in "Scientific Notation." 2.E+10 means a 2 with 10 zeros after it.)





PROBLEM 33 MATH TEST

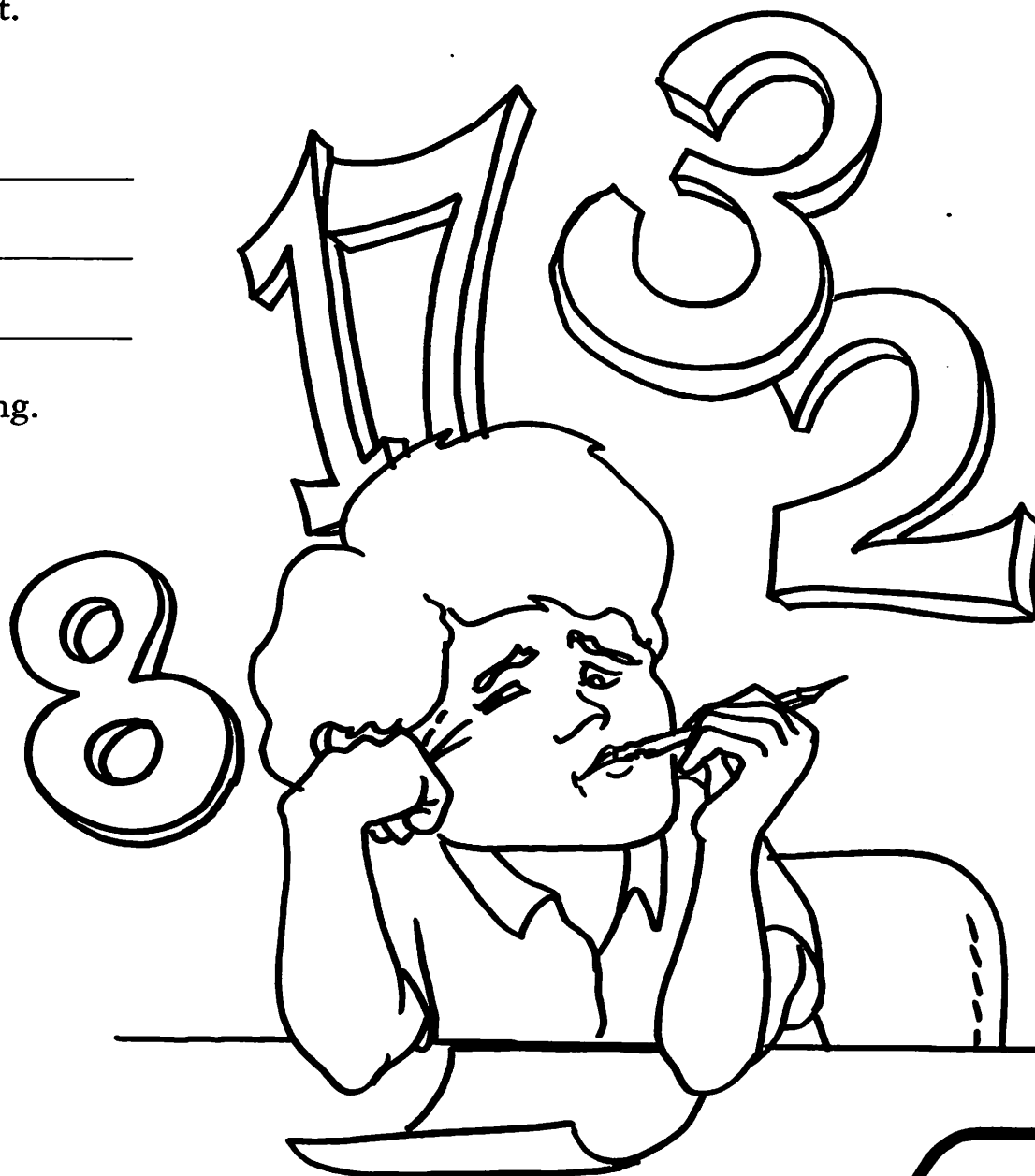
Math quiz time. Run this addition quiz.

```
5 CALL CLEAR
10 PRINT "MATH TEST"
20 PRINT
30 PRINT "TELL ME A NUMBER"
40 INPUT A
50 PRINT "ANOTHER NUMBER"
60 INPUT B
70 PRINT
80 PRINT "WHAT IS"
90 PRINT A;"+";B
100 INPUT C
110 IF C <> A+B THEN 120 ELSE 130
120 PRINT "WRONG"
125 END
130 PRINT "RIGHT"
```

Change this to a multiplication test.

What lines need to be changed?

Make sure your changes are working.



PROBLEM 34 TIMES TABLES

This time we'll make the computer multiply the first ten counting numbers by 2.

Enter and run this program:

```
10 CALL CLEAR
20 PRINT "TIMES TABLES"
30 FOR N = 1 TO 10
40 PRINT N, 2*N
50 NEXT N
```

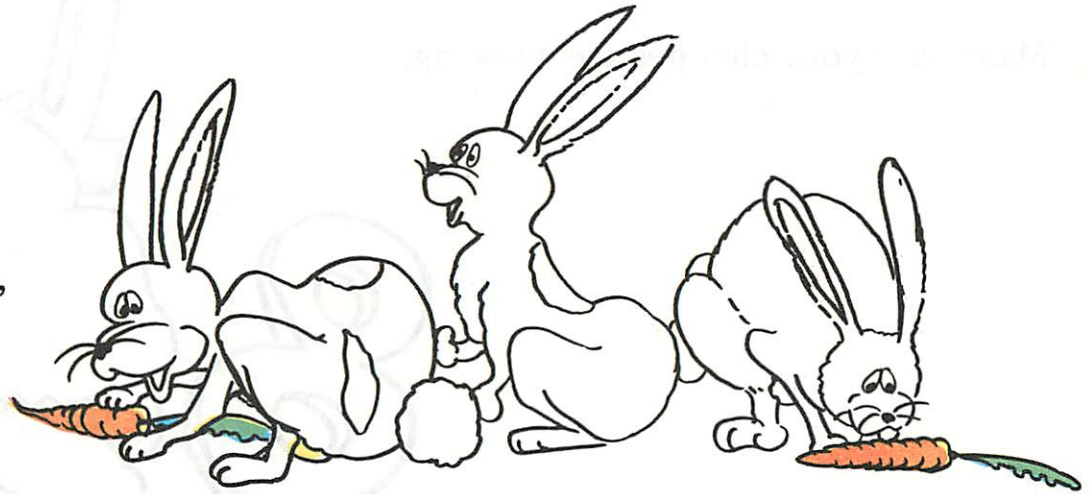
To make the output look like a math book,

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
```

change line 40. We want "2*" and "=" to be printed every time.

```
40 PRINT "2 * ";N;"=";2*N
```

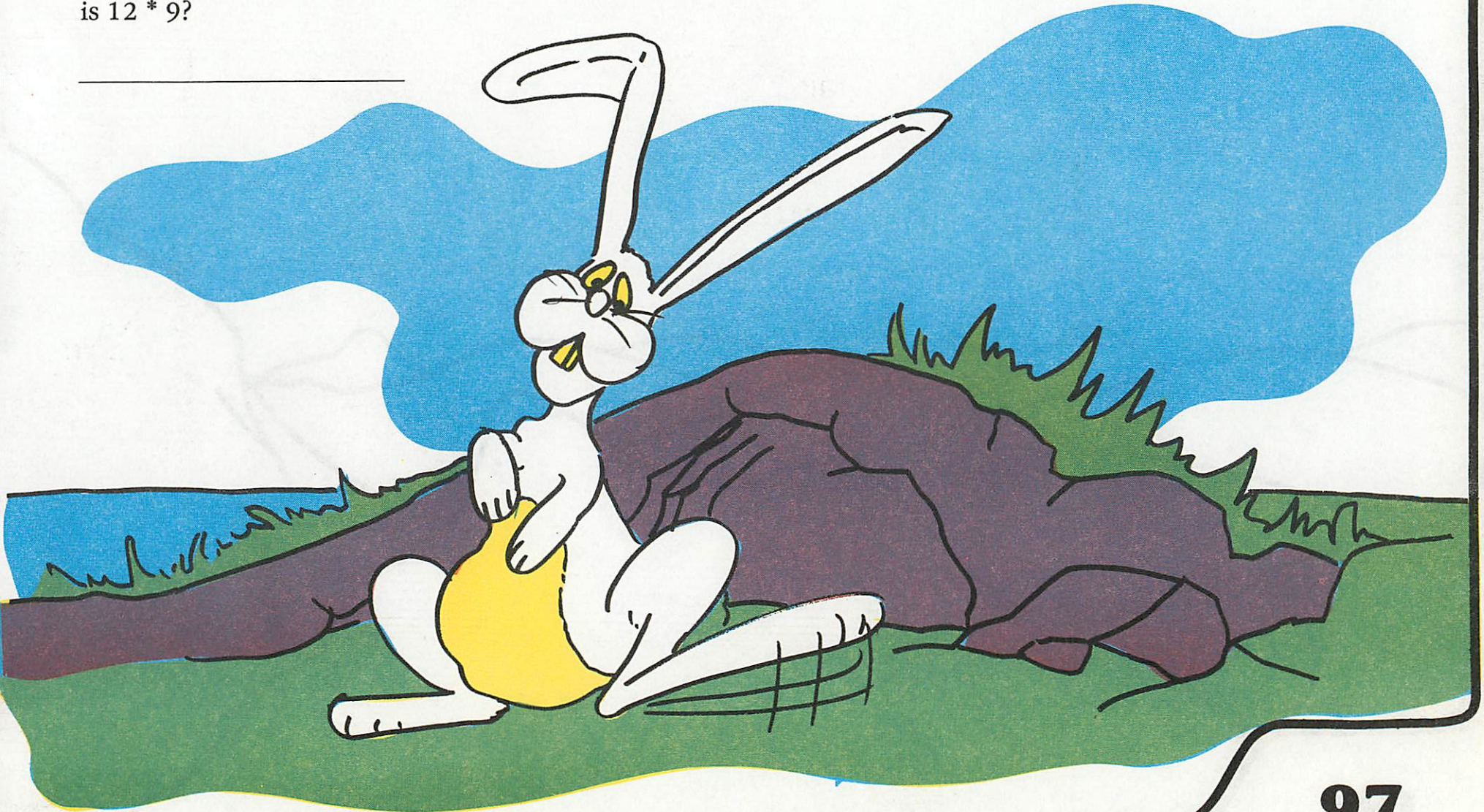
and run the program.



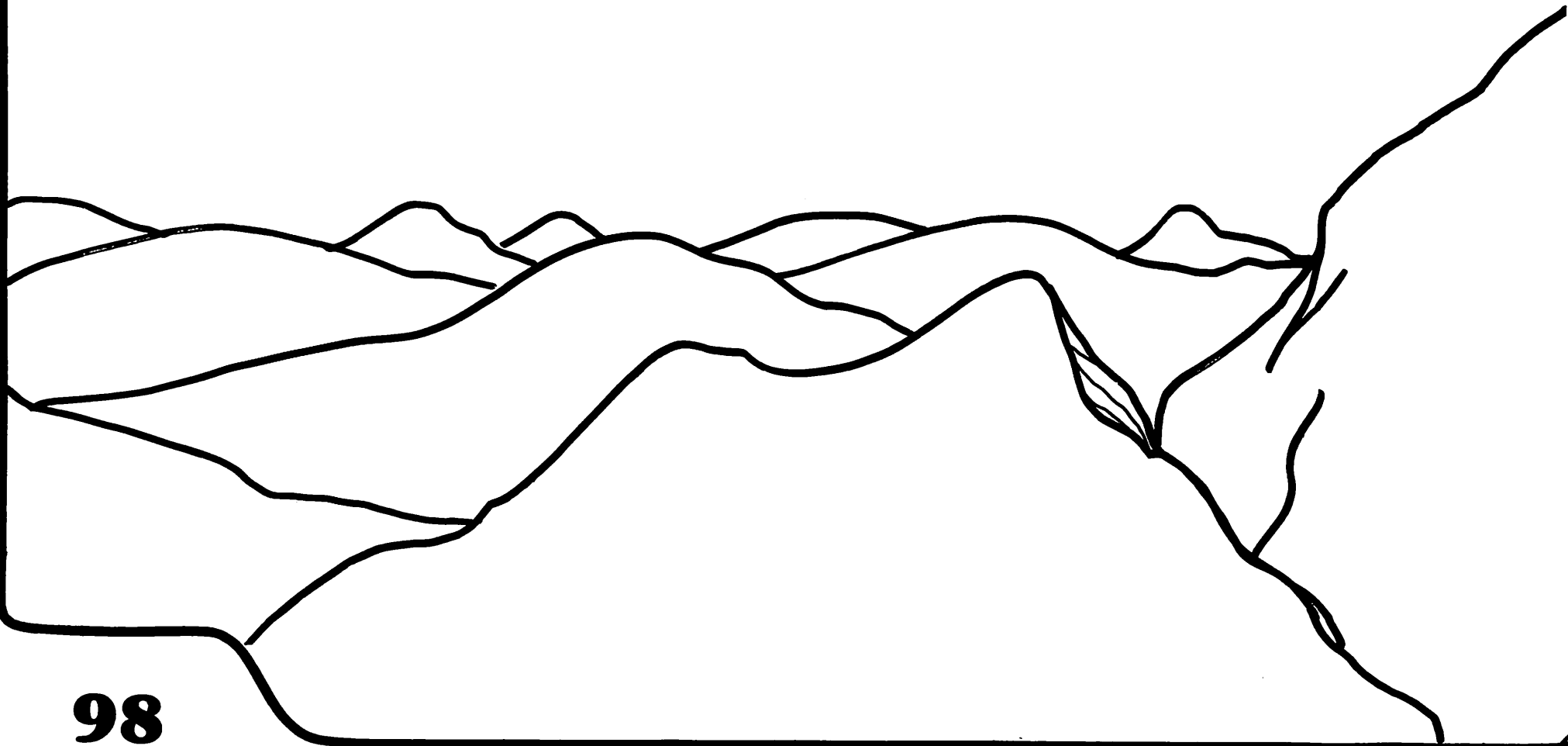
Change line 40 so the computer does the SEVEN's table:

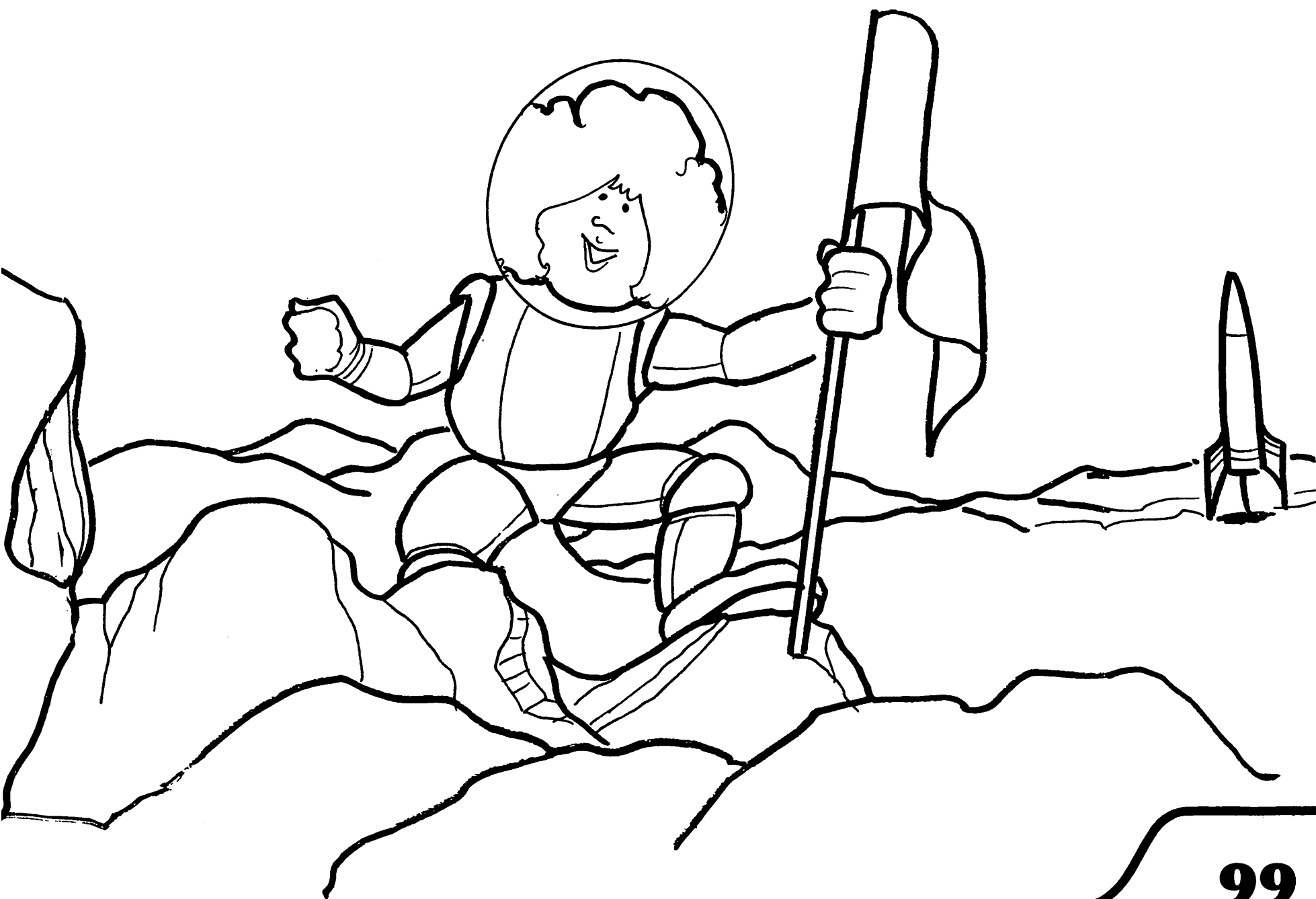
40 _____

Make another change to do the TWELVE's table. How much
is $12 * 9$?



Color me!





PROBLEM 35 DAYS AND NIGHTS

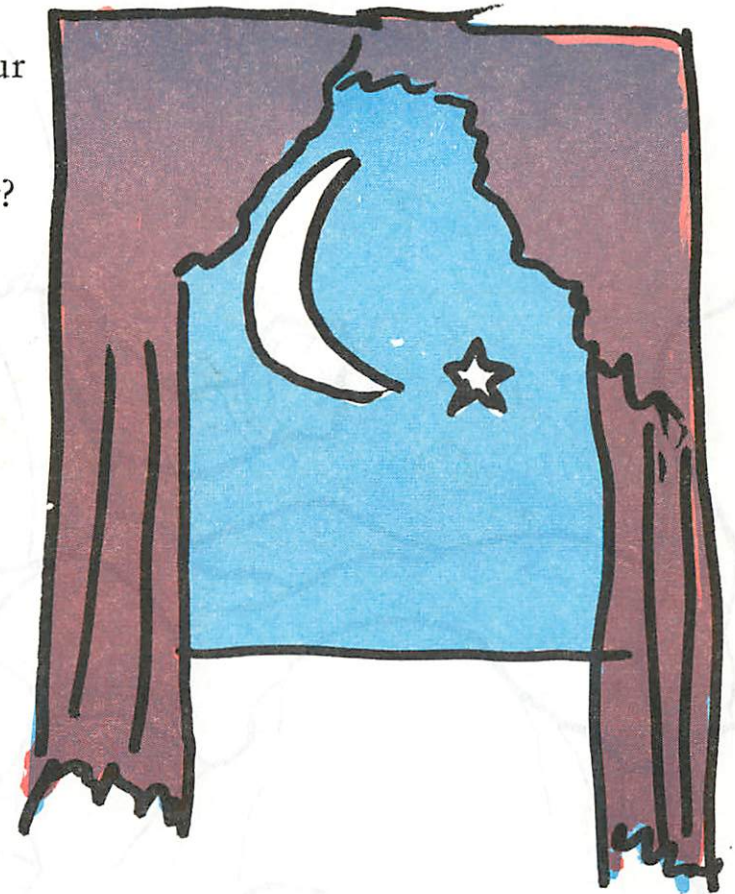
If you spend two hours a day watching television, how many 24-hour days is that equal to?

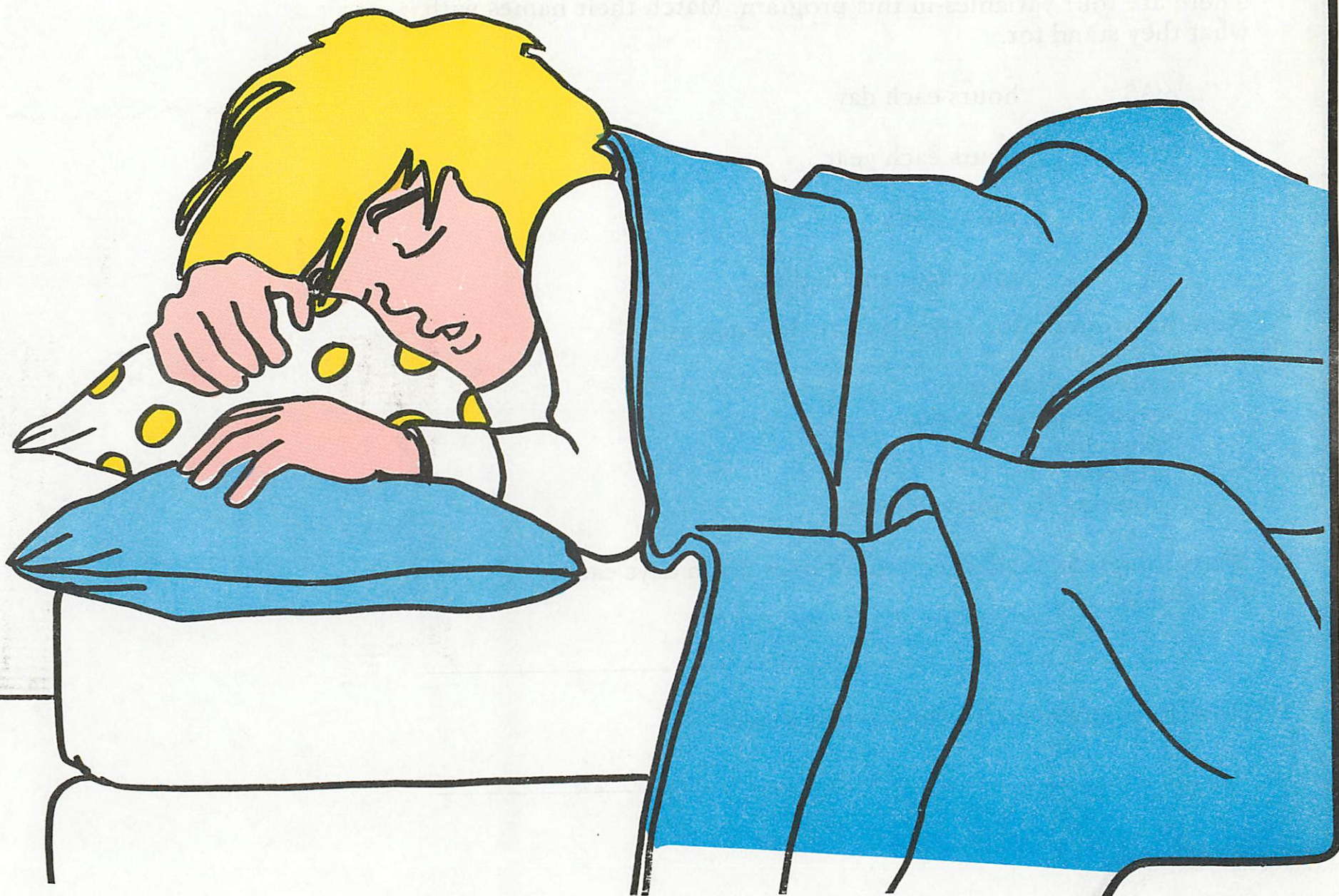
If you sleep eight hours a night, how much sleep is that each year?

The computer can be taught to solve problems like these.

Read this program:

```
100 CALL CLEAR
110 PRINT "WHAT'S THE ACTIVITY?"
120 INPUT A$
130 PRINT "HOW MANY HOURS DO YOU
SPEND EACH DAY"
140 PRINT A$
150 INPUT H
160 HY = H*365
170 PRINT "HOURS EACH YEAR",HY
180 CD = HY/24
190 PRINT "COMPLETE DAYS", CD
```





There are four variables in this program. Match their names with what they stand for.

A\$	hours each day
CD	hours each year
HY	the activity
H	total days and nights

Run the program for activities you do, such as:

- playing the piano
- watching television
- sleeping
- reading
- looking for library books

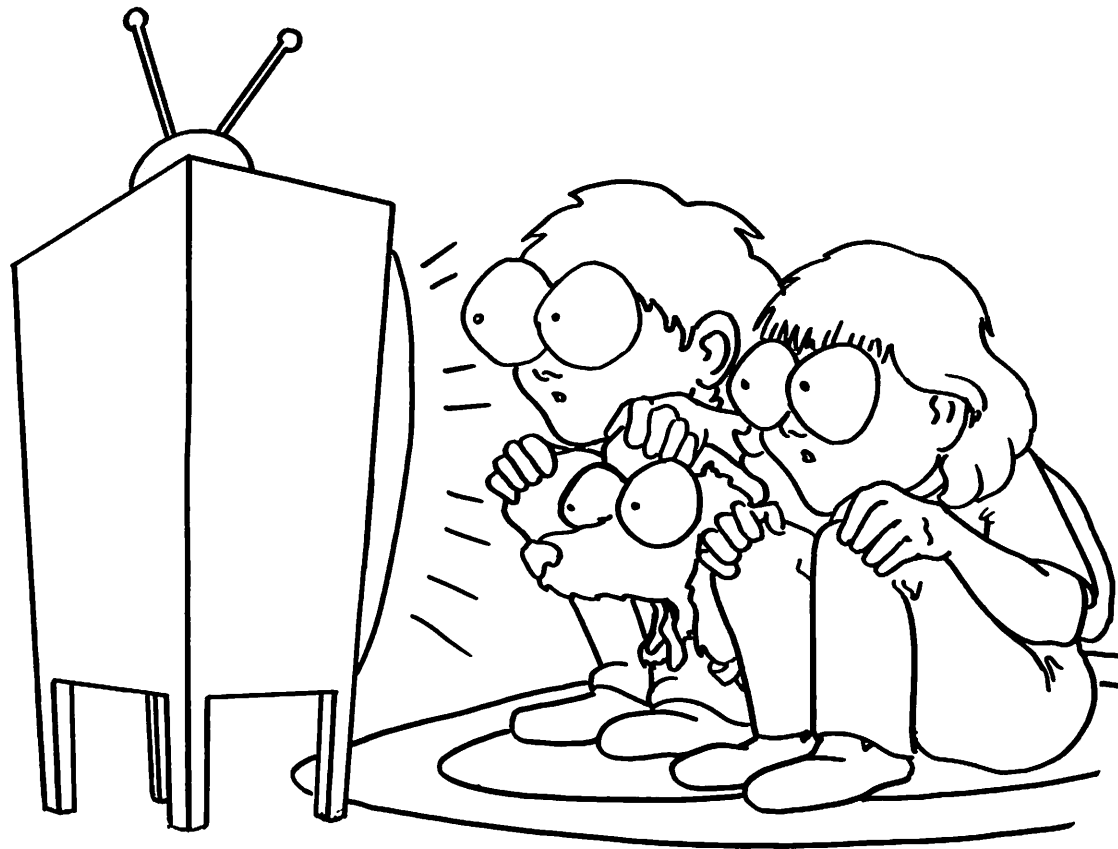
What things do YOU spend more than seven days each year doing?

How do you convert 14 days to two weeks?

Try to add two lines to the program to change CD, the number of complete days, to CW, the number of complete weeks. Print the number of complete weeks.

200 _____

210 _____



PROBLEM 36 COUNTDOWN

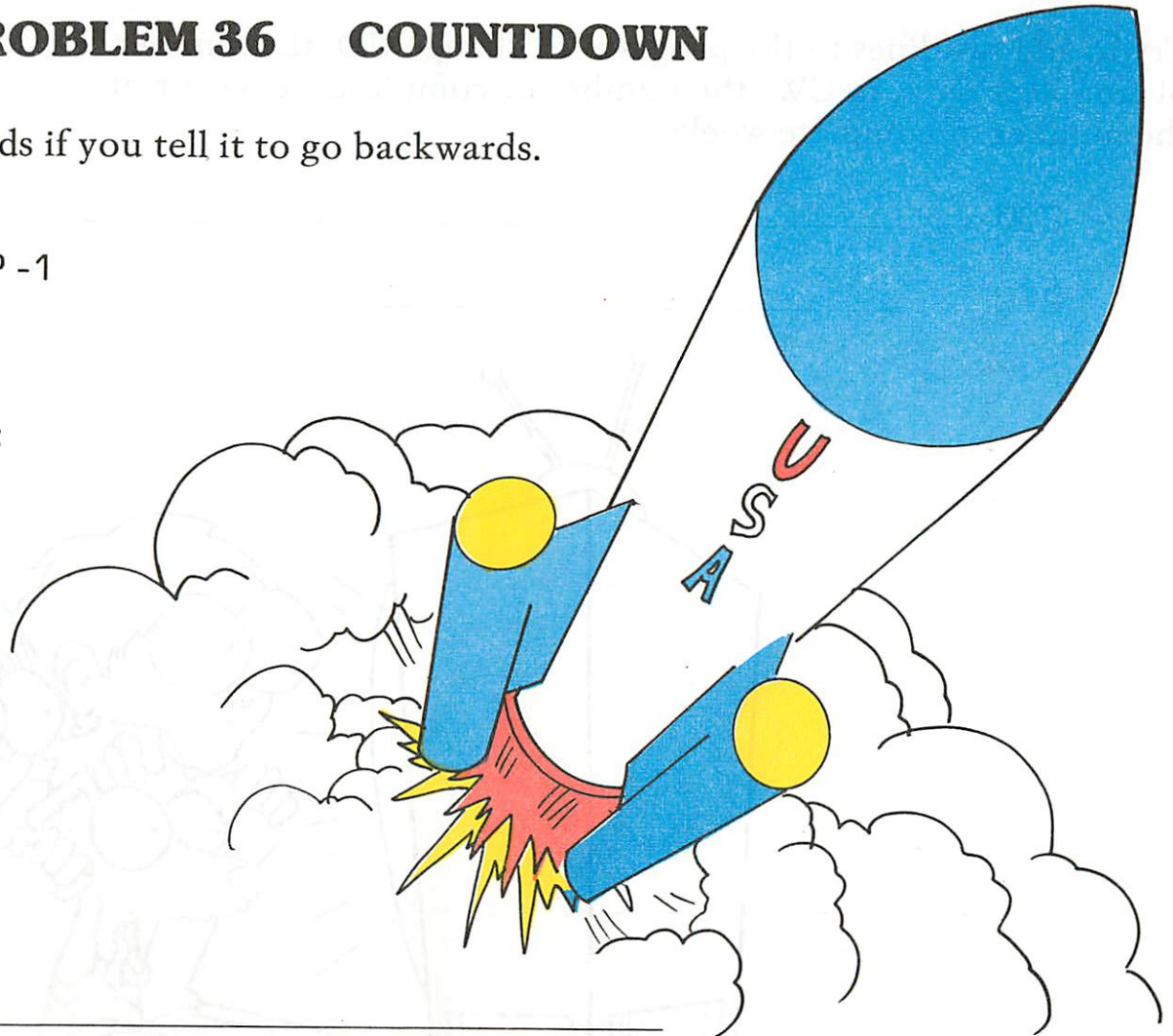
The computer can count backwards if you tell it to go backwards.
Try this program:

```
100 FOR K = 5 TO 1 STEP -1
110 PRINT K
120 NEXT K
```

Write a program which will print:

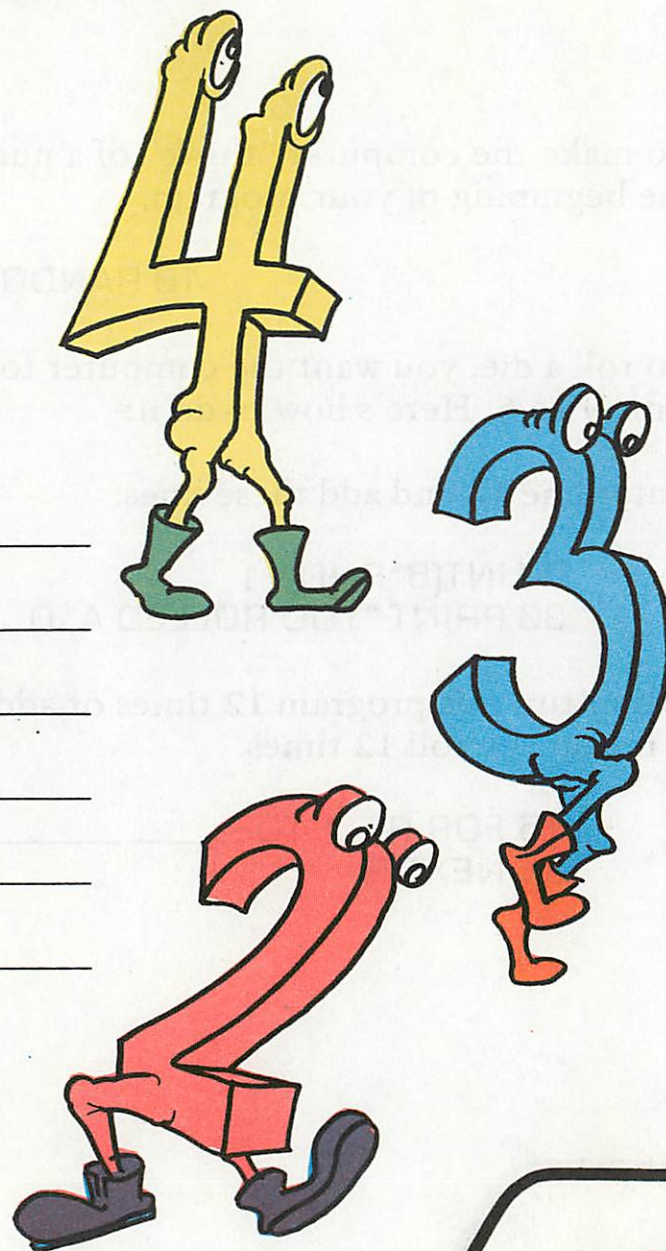
```
10
9
8
.
.
.
1
0
BLASTOFF!
```

Copy your program here.



Write a program which counts from 1 to 5 and then back down to 1 like this:

1
2
3
4
5
4
3
2
1



PROBLEM 37 THINK OF A NUMBER

To make the computer “think” of a number, first put this line at the beginning of your program:

```
10 RANDOMIZE
```

To roll a die, you want the computer to “think” of a number from 1 to 6. Here’s how to do it:

Enter line 10 and add these lines:

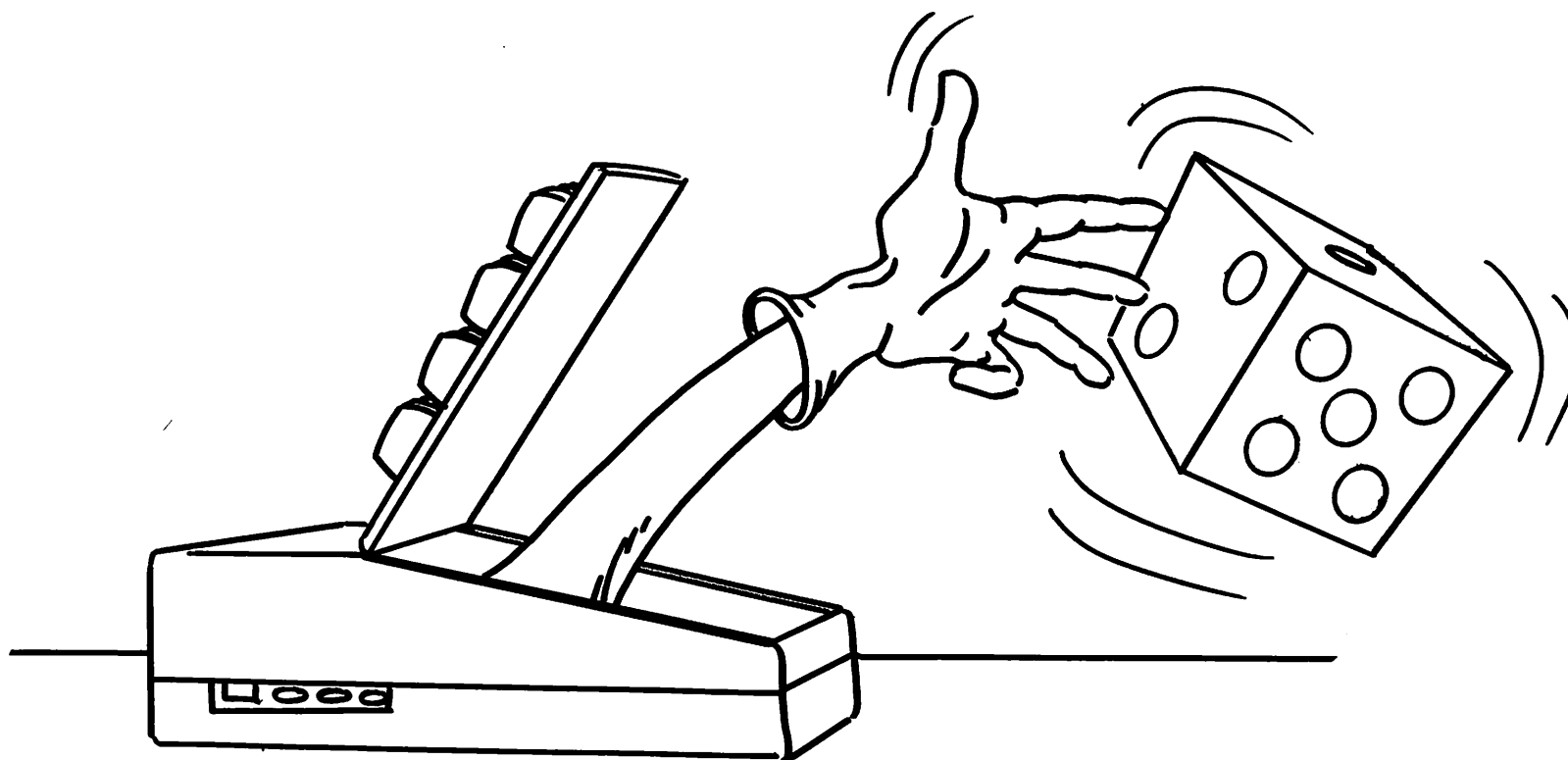
```
20 INT(6*RND)+1  
30 PRINT "YOU ROLLED A",D
```

Either run this program 12 times or add lines 15 and 35 to make it repeat the roll 12 times.

```
15 FOR R = 1 TO _____  
35 NEXT R
```

Record your rolls here:

_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____



PROBLEM 38 TWO DICE

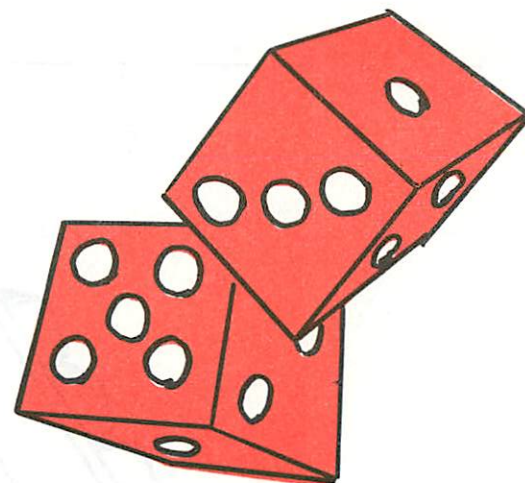
This time we want to roll TWO dice. Each time we roll a die, we must use $\text{INT}(6*\text{RND})+1$.

Enter this program. Either run it 12 times or put in a FOR . . . NEXT loop.

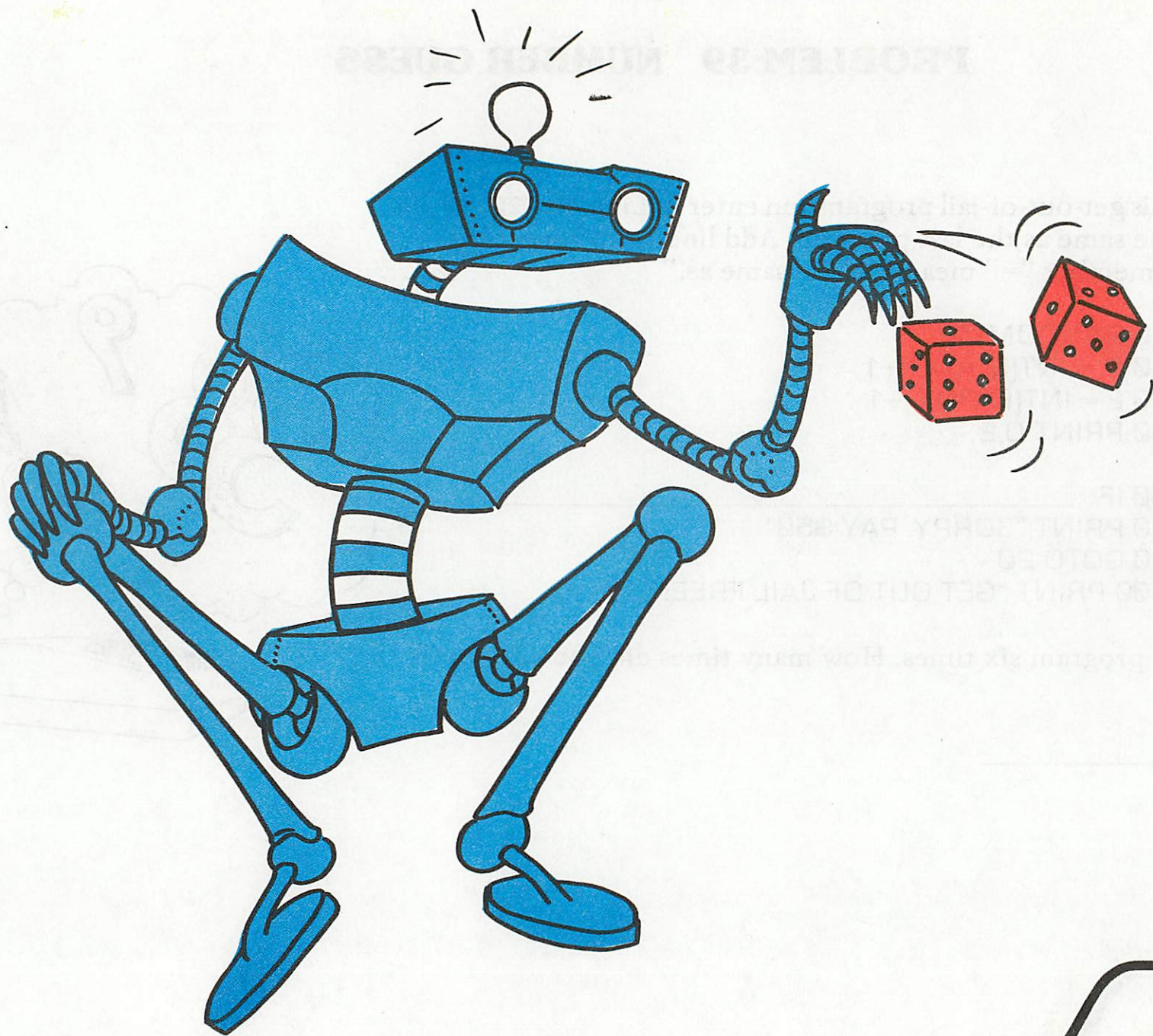
```
10 RANDOMIZE
20 D = INT(6*RND)+1
30 E = INT(6*RND)+1
40 PRINT D,E
```

Record your output. How many times did you get doubles? _____

_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____



Don't NEW this; it's part of the next problem.



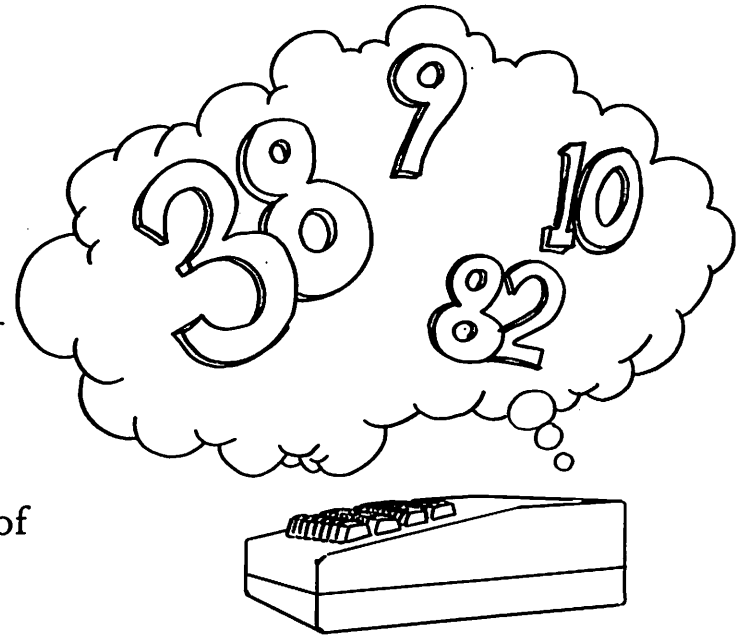
PROBLEM 39 NUMBER GUESS

Finish this get-out-of-jail program and enter it. Lines 10, 20, 30 and 40 are the same as the last problem. Add line 15 to repeat the rolls. Remember "=" means "is the same as."

```
10 RANDOMIZE  
20 D = INT(6*RND)+1  
30 E = INT(6*RND)+1  
40 PRINT D,E
```

```
50 IF _____  
70 PRINT "SORRY. PAY $50"  
80 GOTO 20  
100 PRINT "GET OUT OF JAIL FREE!"
```

Run this program six times. How many times did you get out of jail free?





PROBLEM 40 HARDER MATH QUIZ

RUN this quiz five times.

```
10 RANDOMIZE
20 A = INT(6*RND)+1
30 B = INT(6*RND)+1
40 PRINT A;" + " B;" = ";
50 INPUT C
60 IF C <> A+B THEN 70 ELSE 90
```

```
70 PRINT _____
```

```
80 END
```

```
90 PRINT _____
```



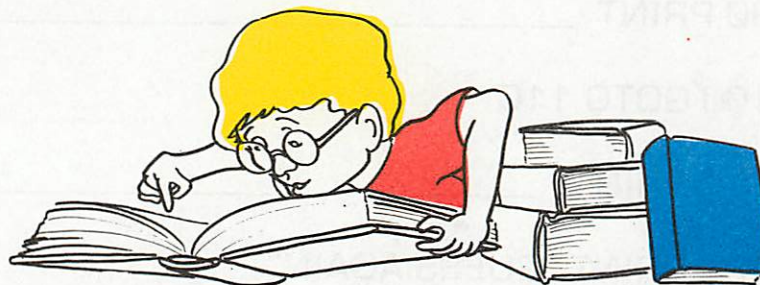
Make one of these changes:

- A. Change it to a multiplication quiz.
What has to be changed?

- B. Let the numbers in the problems get as large as 20.
What has to be changed?

- C. Change line 80 so that if the answer is wrong, the same problem is asked again.

80 _____



PROBLEM 41 NUMBER GUESS

In this program, the computer “thinks” of a number and you try to guess it. If you guess wrong, it tells you TOO BIG or TOO SMALL and asks you to guess again.

Complete the program and run it to check it out.

```
10 RANDOMIZE
20 N = INT(100*RND)+1
30 PRINT "I'M THINKING OF A NUMBER"
40 PRINT "GUESS"
50 INPUT G
```

```
70 IF G = N THEN 500
80 IF G > N THEN 110
```

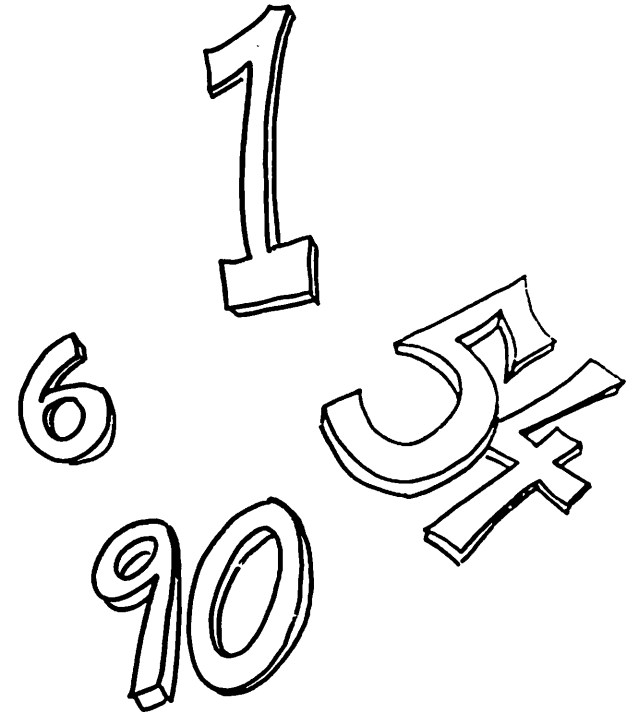
```
90 PRINT _____
```

```
100 GOTO 110
```

```
110 PRINT _____
```

```
120 PRINT "GUESS AGAIN"
130 GOTO 50
```

```
500 PRINT _____
```



What numbers did the computer think of? _____

What's the biggest number it's allowed to think of? _____

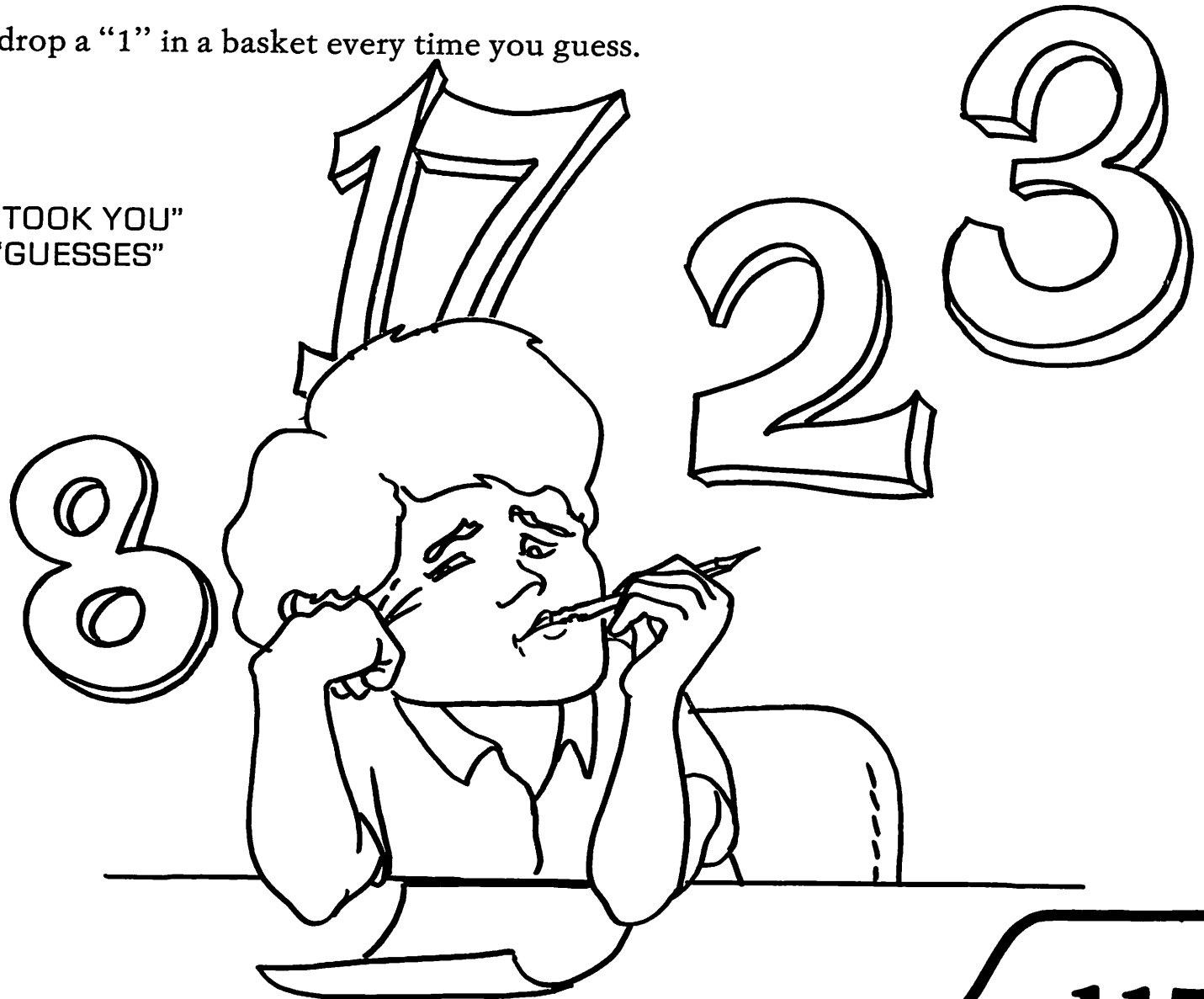
To count the guesses, drop a "1" in a basket every time you guess.
Here's how:

```
5 C = 0
```

```
60 C = C + 1
```

```
510 PRINT "IT TOOK YOU"
```

```
520 PRINT C, "GUESSES"
```



PROBLEM 42

THE ANSWER MACHINE - USING A LIST

Is it going to rain tomorrow? Will I find my watch under the bed?
Will my loose tooth come out today? Are we going to get a pet?

We'll make up a list of ten answers and let the computer pick one each time a question is asked.

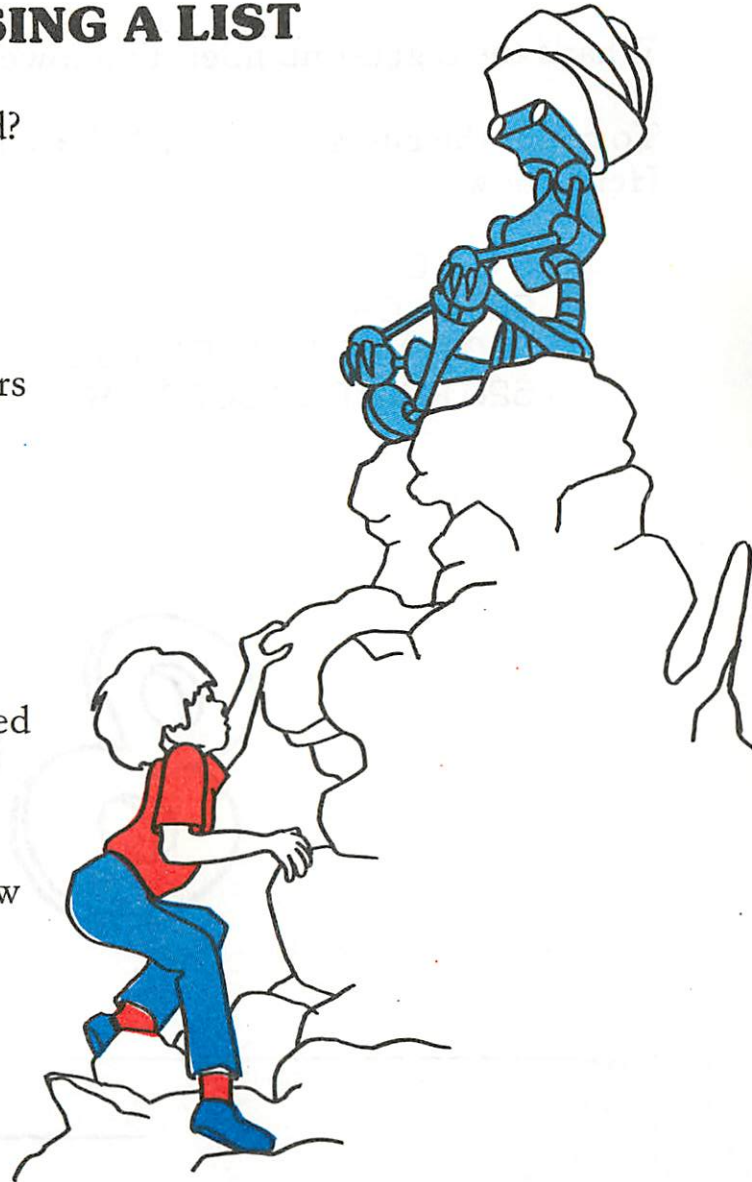
The computer needs a number for every item on a list. The answers in the program will all be called A\$.

The first answer has name A\$(1).
The second answer has name A\$(2).
The third answer has name A\$(3)
and so on.

Whenever a list is used in a program, the computer must be warned how long the list is allowed to get. Line 20 says our list won't have more than 10 things on it.

Finish the program on the next page. Enter and run it to see how a list works. To stop the program you need to use FCTN 4.

```
10 RANDOMIZE 20 DIM A$(10)
21 A$(1) = "YES"
22 A$(2) = "MAYBE"
```



23 A\$(3) = "WHEN DONKEYS FLY"

24 A\$(4) = _____

25 A\$(5) = _____

26 A\$(6) = _____

27 A\$(7) = _____

28 A\$(8) = _____

29 A\$(9) = _____

30 A\$(10) = _____

50 CALL CLEAR

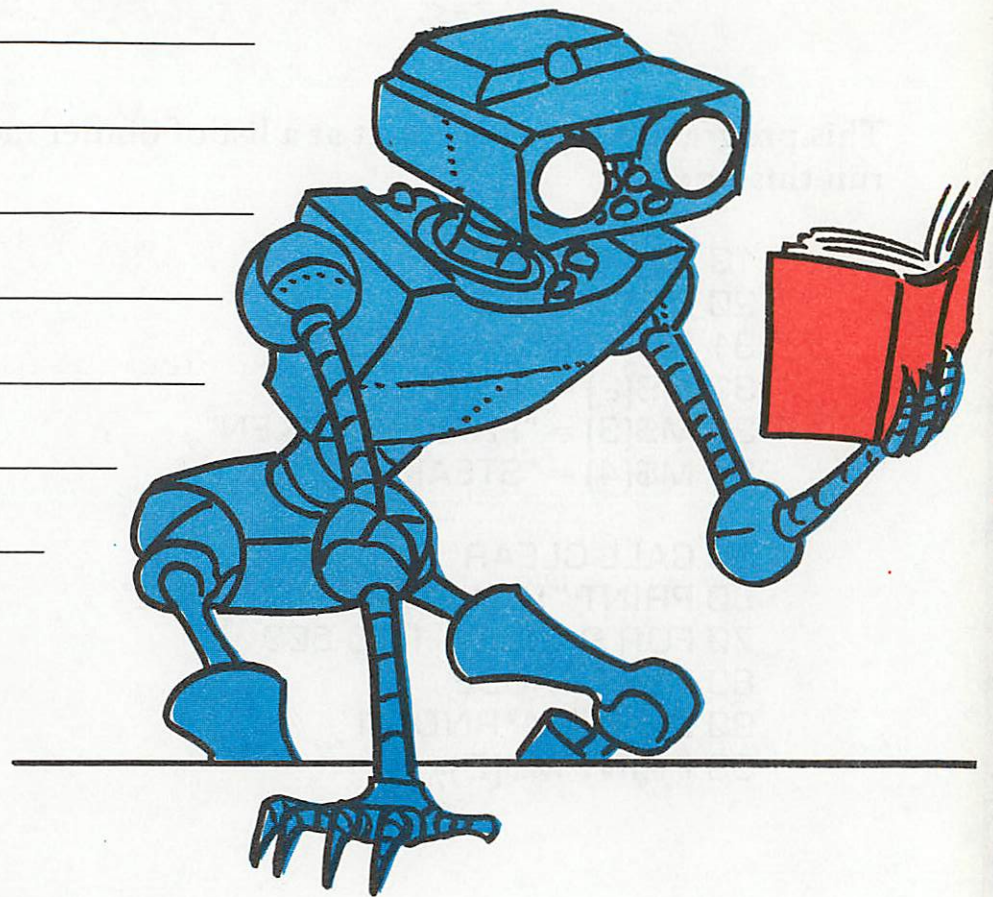
60 PRINT "WHAT IS YOUR QUESTION?"

70 INPUT Q\$

80 N = INT(10*RND)+1

90 PRINT A\$(N)

100 GOTO 60



Put a pause in to make it look as if the computer is thinking about the question.

75 FOR PAUSE _____

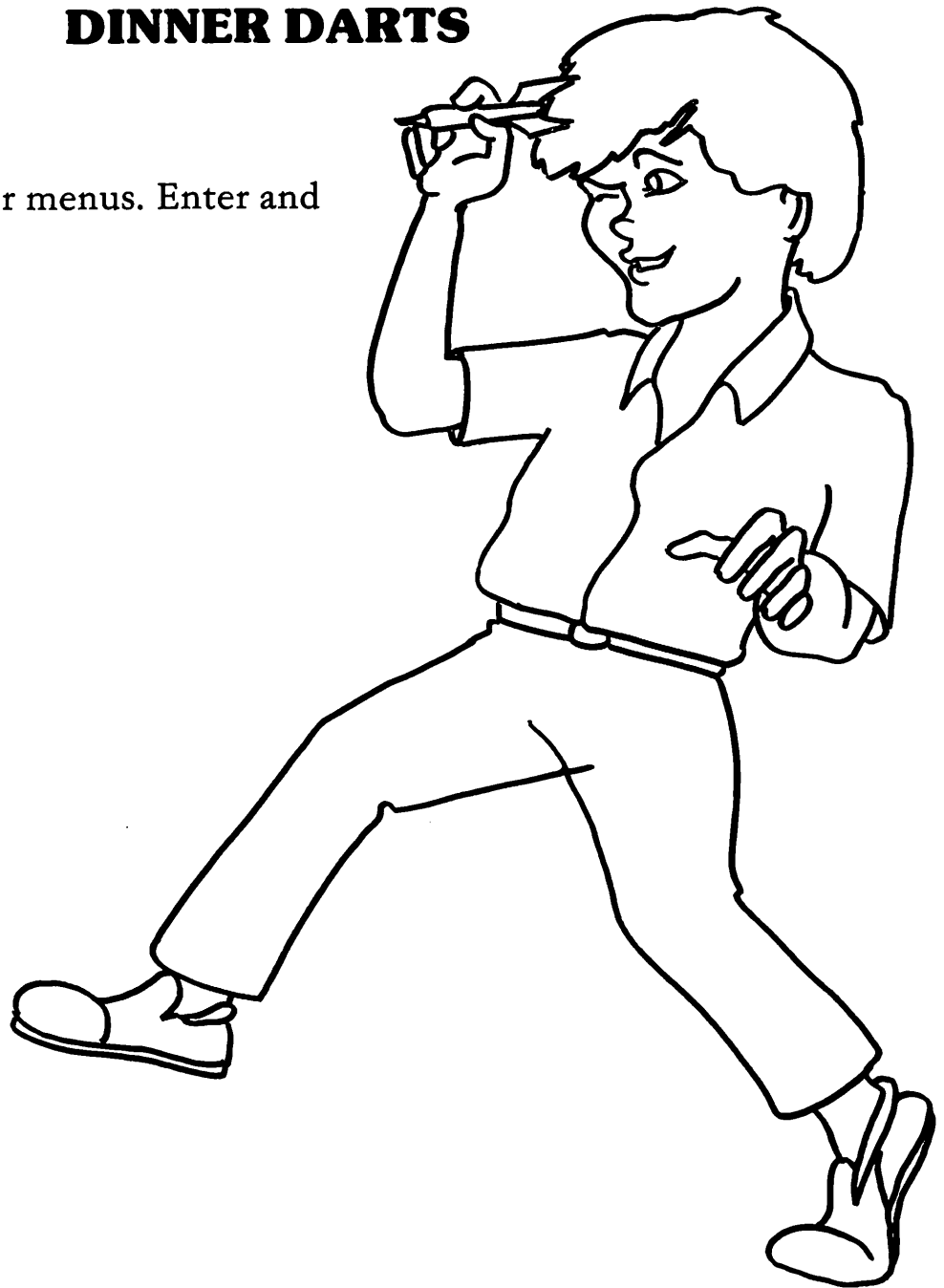
77 NEXT _____

PROBLEM 43 DINNER DARTS

This program will throw a dart at a list of dinner menus. Enter and run this one.

```
10 RANDOMIZE
20 DIM M$(10)
31 M$(1) = "SPAGHETTI"
32 M$(2) = "MEATLOAF"
33 M$(3) = "FRIED CHICKEN"
34 M$(4) = "STEAK"

50 CALL CLEAR
60 PRINT "TONIGHT WE'LL HAVE"
70 FOR PAUSE = 1 TO 500
80 NEXT PAUSE
90 D = INT(4*RND)+1
99 PRINT M$(D)
```



Add four more dishes to the list.

35 M\$(5) = _____

36 M\$(6) = _____

37 M\$(7) = _____

38 M\$(8) = _____

To use your expanded list, change line 90 to

90 D = INT(8*RND)+1



PROBLEM 44 YOUR DAILY HOROSCOPE

The last problem is to write a fortune-telling program. This program has three lists: the kind of day (D\$), what you eat (E\$) and what you play (P\$).

```
10 RANDOMIZE
20 DIM D$(10),D$(10),P$(10)
21 D$(1) = "LUCKY"
22 D$(2) = "SAD"
23 D$(3) = "COLD"
31 E$(1) = "SPAGHETTI"
32 E$(2) = "ICE CREAM"
33 E$(3) = "GRANOLA"
41 P$(1) = "GIN RUMMY"
42 P$(2) = "THE PIANO"
70 CALL CLEAR
75 PRINT "WHAT'S YOUR NAME?"
80 INPUT N$
85 PRINT "TODAY YOU WILL BE"
90 D = INT(3*RND)+1
95 PRINT D$(D)
100 PRINT "ESPECIALLY IF YOU EAT"
105 E = INT(3*RND)+1
110 PRINT E$(E)
```




```
115 PRINT "AND PLAY"
```

```
120 P = INT(2*RND)+1
```

```
125 PRINT P$(P)
```

Now add some more possibilities:

24 D\$(4) = _____

25 D\$(5) = _____

26 D\$(6) = _____

34 E\$(4) = _____

35 E\$(5) = _____

43 P\$(3) = _____

44 P\$(4) = _____

To make sure the crystal ball can see the new possibilities, too,
change the pick-a-number lines:

90 D = _____

105 E = _____

120 P = _____



GLOSSARY

PRINT "HELLO"	puts HELLO on the screen
GOTO 5Ø	sends computer to do line 5Ø next
INPUT N	program stops until the user types a number and presses RETURN
INPUT A\$	program stops until the user types a word and presses RETURN
FOR PAUSE = 1 TO 5 NEXT PAUSE	computer pauses while it counts to 5 (to itself)
FOR K = 1 TO 5 PRINT "HELLO" NEXT K	prints "HELLO" five times
END	program stops here
IF A = 6 THEN ... IF A <> 6 THEN ...ELSE ... IF A\$ = "HORSE" THEN ...	testing whether things are the same or different
CALL KEY(Ø,K,S)	X is the code number of the last key touched.

FOR K = 5 TO 1 STEP -1
PRINT K
NEXT K

counts backwards and prints 5,
4, 3, 2, 1 in a column

D = INT(6*RND)+1)

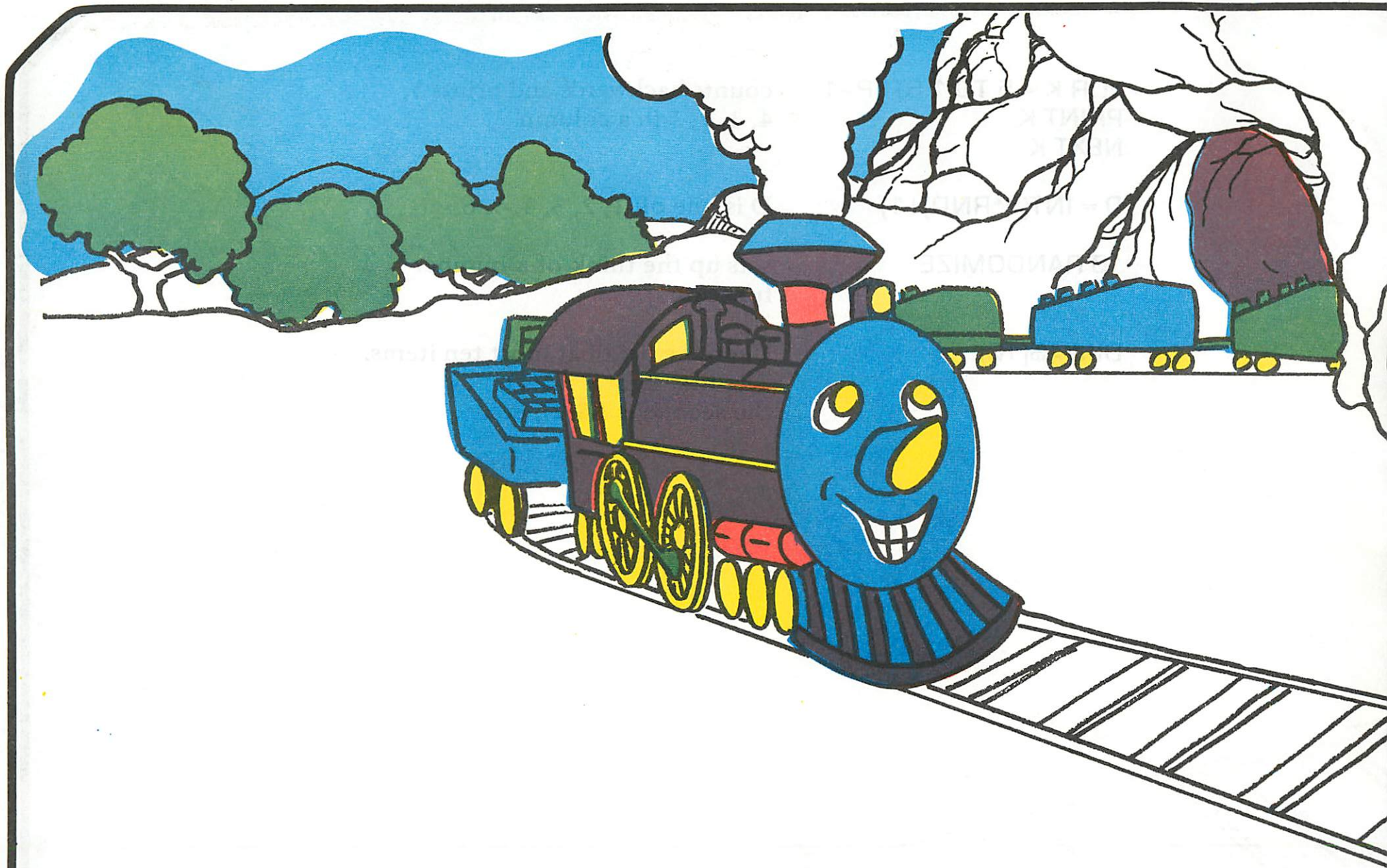
D is one of: 1, 2, 3, 4, 5, 6

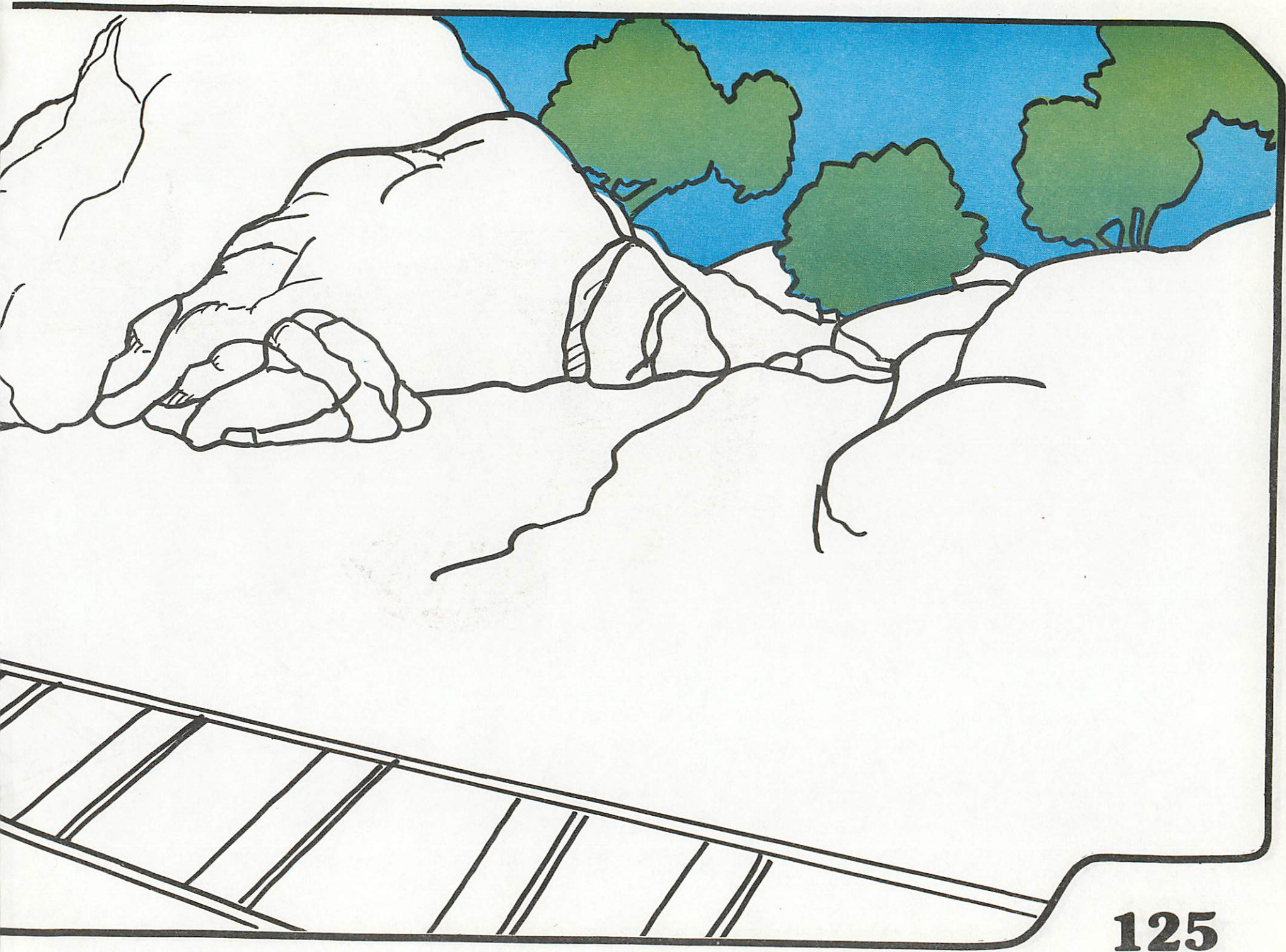
10 RANDOMIZE

sets up the think-of-a-number
function

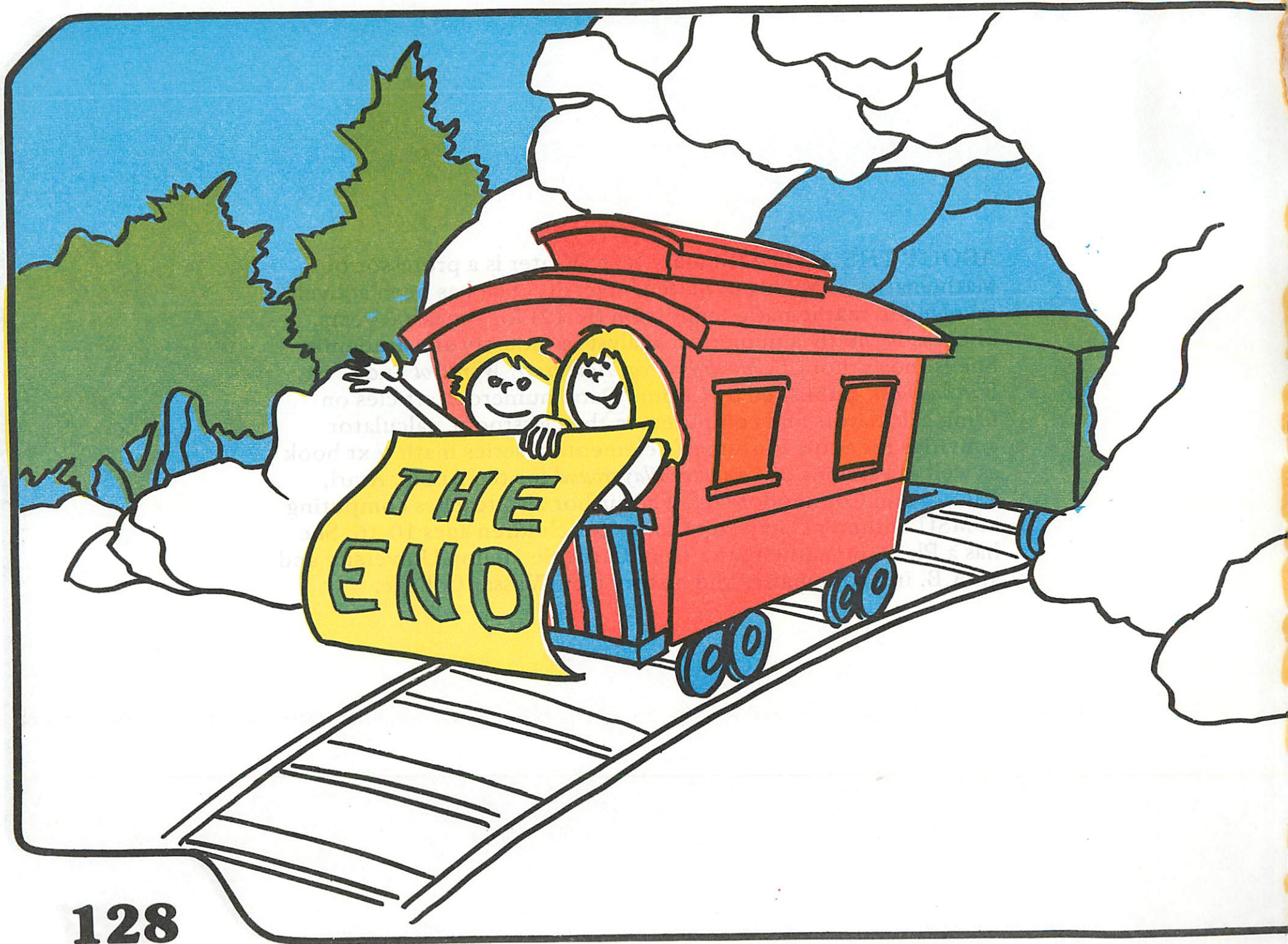
DIM A\$(10)

There's a list of at most ten items.
The first item is A\$(1),
the second is A\$(2), etc.





ABOUT THE AUTHOR: Mary Jean Winter is a professor of Mathematics at Michigan State University. She has been actively involved in mathematics education (K-12) for the last 8 years. She was formerly a numerical analyst in industry and academia. She is the author of *Chivalry*, *Great Adventure*, *Wordspot* and *Witchnumber*, published by Comm*Data; numerous articles on using calculators and computers in the classroom, calculator activities for Scott Foresman's elementary series math text book as well as versions of *Computer Playground* for the Apple, Atari, VIC-20 and Commodore 64. The author also teaches computing at MSU Summer Computer Camps for children ages 10-16. She has a Ph.D. in mathematics from Carnegie Mellon University and an A.B. in mathematics and history from Vassar College.



COMPUTER ACTIVITIES FOR THE YOUNG CHILD

COMPUTER PLAYGROUND, by Mary Jean Winter, is based on children's natural interest in words, games and graphics. A collection of BASIC computer activities for the TI 99/4A, this book is intended for children in grades 2 through 6 who have access to a computer.

Each activity is presented as a "problem" with questions to lead the child to the correct answer. Children will find the problems interesting, fun and most importantly — geared to their level of understanding. They'll run sample programs, learn how to modify them, and complete partially written programs. Everything is kept simple — the child is led step-by-step through the programming process.

As the child progresses, he or she will learn how to use many of the features of BASIC — PRINT, GOTO, INPUT, FOR . . . NEXT, IF . . . THEN, and CALL commands are a few. By working in this manner, the child will build up a collection of TI BASIC programs which are both understandable and entertaining!

COMPUTER PLAYGROUND is illustrated with over 70 color cartoons and pictures to color. It's BASIC programming playtime!

COMPUTER PLAYGROUND is also available for the Commodore 64/VIC-20, Atari 400/800/1200, Apple II, II+ and //e computers.

OTHER POPULAR COMPUTER BOOKS BY DATAMOST

Kids & the Apple
Kids & the Atari
Kids & the Commodore 64
Kids & the Panasonic
Kids & the TI 99/4A
Kids & the VIC-20
by Ed Carlson

How to Write an Apple Program
How to Write an IBM-PC Program
How to Write a TRS-80 Program
How to Write a Program Vol. II
by Ed Faulk

Games Apples Play
by Mike Weinstock
& Mark Capella

The Elementary Apple
The Elementary Atari
The Elementary Commodore 64
The Elementary TI 99/4A
The Elementary Timex/Sinclair
The Elementary VIC-20
by William Sanders

Computer Playground Apple
Computer Playground Atari
Computer Playground Commodore 64/VIC-20
by M. J. Winter

Using 6502 Assembly Language
p-Source
by Randy Hyde



8943 Fullbright Avenue
Chatsworth, CA 91311-2750
(213) 709-1202

ISBN 0-88190-297-7

